DATA REDUCTION SYSTEM FOR THE XR-3 CAPTURED AIR BUBBLE TESTCRAFT

Peter Vincent Dabbieri

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

DATA REDUCTION SYSTEM
FOR THE XR-3
CAPTURED AIR BUBBLE TESTCRAFT

by

Peter Vincent Dabbieri, Jr.

Thesis Advisor: D. M. Layton

December 1973

Approved for public release; distribution unlimited.

Data Reduction System
for the XR-3
Captured Air Bubble Testcraft

by

Peter Vincent Dabbieri, Jr.
Lieutenant, United States Navy
B.S., United States Naval Academy, 1967

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1973

ABSTRACT

A fast, accurate and portable data reduction system was developed for the XR-3 Captured Air Bubble testcraft being evaluated at the Naval Postgraduate School, Monterey, California. The system consists of four units. A magnetic tape recorder is used for data play back. A signal conditioner unit with a built-in analog-to-digital converter was developed and is used to filter, amplify, sum and further prepare the data for transmission to either a strip chart recorder or a digital X-Y plotter through a Monroe 1880 calculator.

Preliminary use of curve fitting techniques are discussed; calculator programming and the various problems and solutions encountered in the development of the system are described.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

TABLE OF ABBREVIATIONS

ac	alternating current
A/D	analog-to-digital
BCD	binary coded decimal
BYIN	byte input
BYOT	byte output
CAB	captured air bubble
DC	(data) address register
dc	direct current
DH	high (left) six bits of DC register
DL	low (right) eight bits of DC register
DTEN	data transfer enable
E	entry register
FM	frequency modulated
Hz	hertz
IH	high (left) four bits of IX register
IL	low (right) four bits of IX register
in	inch
I/O	input and/or output
IPDB	input data byte
IX	index register
MDR	main data register
mV	millivolt
NPS	Naval Postgraduate School, Monterey, California
OPCB	output control byte
OPDB	output data byte
OTCT	output control time
RAM	random access memory
RINS	read input status into IX register
ROM	read only memory
s	second
SOR	successive over-relaxation
V	volt

ACKNOWLEDGEMENT

The author wishes to thank Professor Donald M. Layton for originally suggesting the problem and for his continuous encouragement and guidance throughout the project. He also wishes to thank Mr. Theordore Dunton and Mr. Michael Odell for both their technical assistance and their help in acquiring the materials needed to complete the project; and Mr. James Sturges for his help in the designing and building of logic circuitry.

I. INTRODUCTION

Associated with most experimental research projects is the requirement to obtain, evaluate, interpret, and analyze large quantities of data in order to draw meaningful conclusions. A manual data reduction method is both time consuming and prone to error. On the other hand, an automatic data reduction system is more accurate and can significantly shorten the man-hours required to reduce raw experimental data into a form which can be quickly utilized by the project engineers.

An automatic data reduction system was needed for the XR-3 Captured Air Bubble (CAB) testcraft being evaluated by the Naval Postgraduate School (NPS), Monterey, California. This craft is being operated at Lake San Antonio, approximately 100 miles from the school. Test data are recorded on a 14-track Pemco, Model 120B, magnetic tape recorder. The method used to reduce the data has consisted of transporting the tape unit back to NPS, using a digital voltmeter and/or a strip chart recorder to obtain a visual data presentation, and then making hand performance plots. The ratio of data acquisition time to reduction time was about 1 to 5. The need for a faster system arose from the decision to acquire a facility that would permit the test operators to remain overnight at the test site. In order to maximize the efficiency of

operating on-site over a several-day period, the results of each day's testing must be reduced and analyzed before a schedule of tests can be prepared for subsequent test runs. Therefore, the time to reduce the data must be shortened significantly.

The automatic data reduction system described in this paper fulfills these objectives. It provides for a better utilization of operating time at the test site by providing a quick, accurate and portable means of evaluating the performance of the testcraft.

II. BACKGROUND

A. NATURE OF THE PROBLEM

Both the data acquisition system and the data reduction methods for the XR-3 have been constantly improved since the testcraft arrived at the Postgraduate School in March, 1970. Initially, performance data were limited to thrust, velocity and plenum cavity pressure. These parameters were hand recorded and hand reduced to produce rough studies of thrust versus velocity and optimum plenum pressure. However, in March, 1973, a new data acquisition system was installed [Ref. 1] in order to test and evaluate the performance of the XR-3 with respect to various other operating parameters such as bow and stern seal pressure, turning speeds, turning rates and roll damping. This is the present system and the one for which the automatic data reduction system was to be designed and developed. A brief description of the data acquisition system can be found in Appendix A.

The purpose of the new automatic data reduction system was to provide a quick and efficient means of transforming raw tape-recorded performance data from tests of the XR-3 into a finished graphical presentation capable of being used for evaluating the effects of varying operating parameters. The system was to have the capability of producing a printed record of the data converted into physical

engineering units, such as knots for velocity vice the millivolts (mV) output from the tape recorder. The system had to be portable, in order to be moved in and out of the data reduction facility, and also small and capable of being operated in a limited space. The equipment had to be simple and easy to use, requiring few inputs from the operator. One of the major considerations was that the system had to be expandable, which would allow for the implementation of future requirements.

The presentation of the results was considered an important factor. Three-digit data accuracy was sufficient both for plotting and for the printed output since the raw data are accurate only to this order of magnitude. The data would require being scaled twice, once for the plot, changing millivolts to inches on the graph, and once for printing, if desired, changing millivolts to physical units. The system also had to be capable of performing operations such as summing inputs from two or more channels of the tape recorder. One such application is in determing the total drag on the testcraft at a constant speed. In this condition the drag equals the thrust, which in turn equals the sum of the thrust from both the port and the starboard engines.

All of the phenomena being tested occur at a fairly low frequency, no greater than about 0.5 Hz. For this reason the sampling rate need not be much greater than

one Hz according to Shannon's sampling theorem (also known as Kirchhoff's sampling theorem). However, to allow for future expansion a faster sampling rate would be needed.

B. MEETING THE OBJECTIVES

A Monroe model 1880 scientific programmable printing calculator and a Monroe model PL4 digital plotter were chosen to be the basic components of the new data reduction system (see Figure 1). Because of its small size and relatively large instruction set, the calculator seemed best suited to perform any manipulations necessary to prepare the data for the plotter and also to provide a paper tape printout of the data converted to the proper physical units.

An analog-to-digital (A/D) converter was necessary to transform the analog tape recorder signal to a digital form compatible to calculator input. A contract was let to Santa Cruz Engineering of Santa Cruz, California in April, 1973 to design and build an interface which would make it possible for the calculator to select a tape recorder channel, then convert the signal from that channel to a digital input to the calculator. Calculator software (program) would convert this input into the various forms necessary to accomplish the remaining tasks. A delivery date of June 1973 was set; however, an extension was required until late July 1973.

As an intermediate method of data reduction while waiting for the interface, various methods of curve fitting

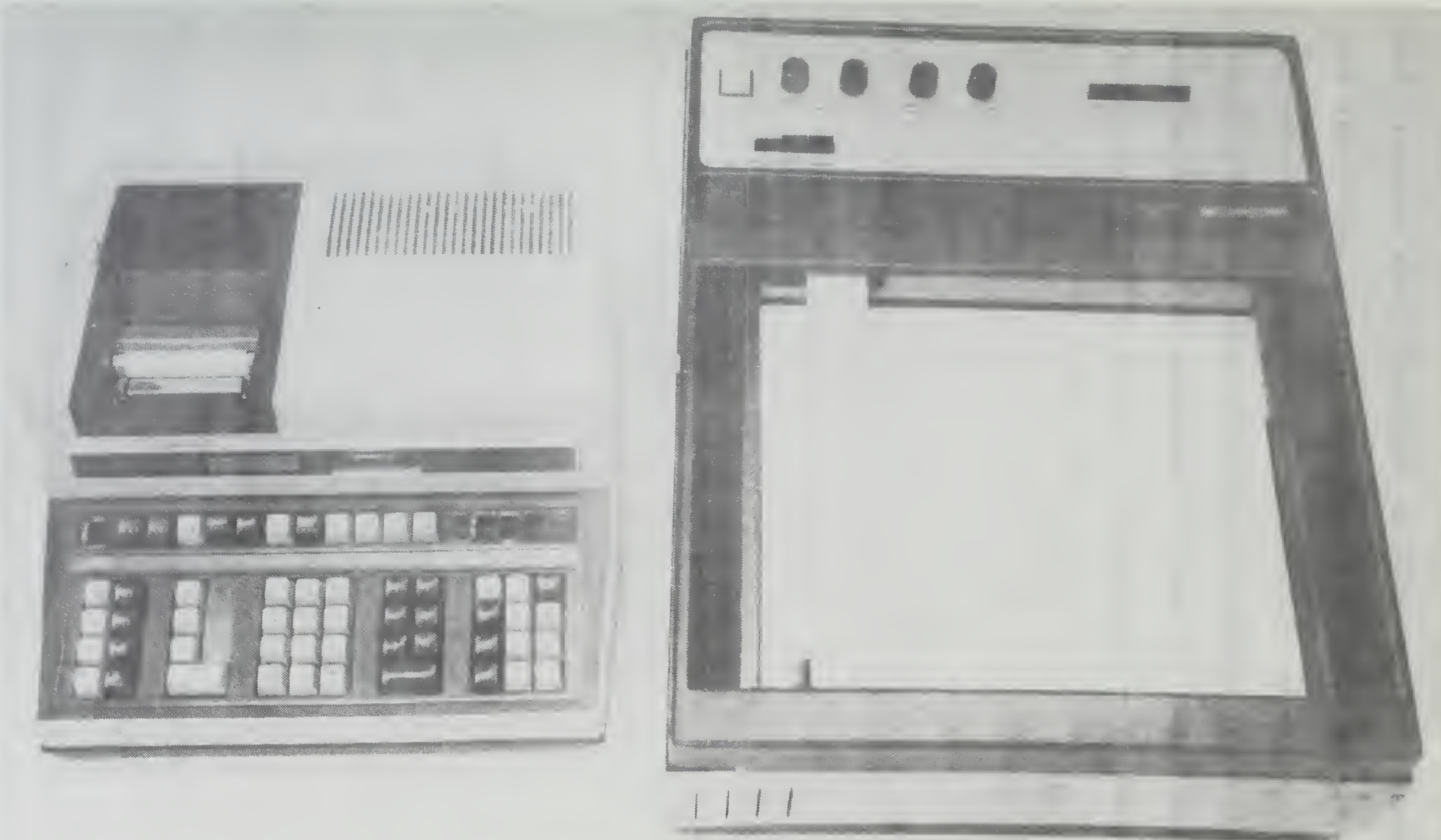


Figure 1. Monroe 1880 Calculator and PL4 Plotter.

and plotting were developed on both the calculator and on the IBM 360/67 digital computer available at NPS. These methods are discussed in Section III. Because the calculator solutions were temporary in nature and presented many problems, the solutions to some of which were not obtained, only the significant portions will be discussed.

III. CURVE FITTING

A. GENERAL METHODS

There are various methods available to engineers for fitting data to an algebraic relation that lends itself to easy manipulation and plotting and that can have a physical significance. The methods explored for application to the data reduction system were the following:

1. Piecewise - polynomial interpolation method
2. Difference method
3. Least-squares approximation by orthogonal polynomials method
4. Least-squares regression method

1. Piecewise-Polynomial Interpolation Method

The cubic spline method of piecewise-polynomial interpolation was not satisfactory for use by either the calculator or the IBM 360 computer for two reasons. First, the method required knowledge of the slope of the curve at the initial and final points. This information is not known and only a gross estimate could be achieved. Secondly, a spline fit requires the curve to pass through each of the data points. Recording error and signal noise are not smoothed by this method and as a result an inaccurate model of the physical event results. A spline fit is best suited to determine polynomial approximations for exact data whose slope has discontinuities [Ref. 2].

2. Difference Method

The difference method likewise does not lend itself to computer application; however, it is an excellent tool for determining the degree of fit required for a least-squares type method. When the difference method is employed, the data must be gathered such that the dependent variable is obtained at equal increments of the independent variable (X). A difference table is formed taking iterations until an essentially constant difference is reached throughout a given iteration. The order of the polynomial to give the best fit is the same as the number of iterations to reach this constant difference [Ref. 3]. That is, if the second iterations produce a constant value, then the model will be of the form $y = a_0 + a_1 x = a_2 x^2$. The coefficients of the polynomials are solved for by using the Gregory-Newton interpolation formula:

$$y = y_o + r(\Delta y_o) + \frac{r(r-1)(\Delta^2 y_o)}{2!} + \frac{r(r-1)(r-2)(\Delta^3 y_1)}{3!}$$

$$+ ------ + \frac{r(r-1),---,(r-n)(\Delta^n y_o)}{n!}$$

where

$$r = \frac{x - x_o}{w}$$

w is the increment between successive values of the independent variable x, and

y_o, x_o are the initial values of the variables

EXAMPLE 1 DIFFERENCE METHOD

As an example of this method, the data for a third order equation $y = 5 + 2x^3$ are presented in Table I. Since the constant difference occurs in the third iteration, the desired fit will be given by a third order polynomial. In addition, since the third iteration resulted in a difference value with zero variance, the fit will be exact.

TABLE I

Successive Differences for Difference Method of Curve Fitting.

Given Data x	y	Successive Differences: 1st Iteration Δy	2nd Iteration $\Delta^2 y$	3rd Iteration $\Delta^3 y$	4th Iteration $\Delta^4 y$
0	5	$\Delta y_0 = 16$	$\Delta^2 y_0 = 96$	$\Delta^3 y_0 = 96$	$\Delta^4 y_0 = 0$
2	21	$\Delta y_1 = 112$	$\Delta^2 y_1 = 192$	$\Delta^3 y_1 = 96$	$\Delta^4 y_1 = 0$
4	133	$\Delta y_2 = 304$	$\Delta^2 y_2 = 288$	$\Delta^3 y_2 = 96$	
6	437	$\Delta y_3 = 592$	$\Delta^2 y_3 = 384$		
8	1029	$\Delta y_4 = 976$			
10	2005				

$$x_0 = 0,\ y_0 = 5,\ w = 2 \quad \therefore\ r = \frac{x-0}{2} = \frac{x}{2}$$

and

$$y = 5 + \frac{x}{2}(16) + \frac{x}{2}\left(\frac{x}{2} - 1\right)\left(\frac{96}{2!}\right) + \frac{x}{2}\left(\frac{x}{2} - 1\right)\left(\frac{x}{2} - 2\right)\left(\frac{96}{3!}\right) + 0$$

$$y = 5 + 2x^3$$

which is the exact equation from which the data were derived. To determine the order of a model from data which is not exact, each column of the difference table must be inspected for an essentially equal distribution of plus and minus signs. The number of iterations to that point determines the order of the resulting polynomial fit. The remaining procedure is as discussed in the preceding example.

3. Least-Squares Approximation by Orthogonal Polynomials

The method of least-squares using orthogonal polynomials could not be used as a calculator solution to the problem since more storage is required by the method than is available in the calculator. However, it does provide a very efficient and accurate solution when used on the IBM 360/67 computer.

The method consists of solving for successively higher order orthogonal polynomials, such that the sum of which satisfies the requirement that

$$\sum_{n=1}^{N} [f(x_n) - p(x_n)]^2$$ where

$f(x)$ = exact solution

$p(x)$ = polynomial approximation

N = order of the polynomial

is as small as possible [Ref. 2]. A computer program was developed for use on the IBM 360/67 in which the X and y data points, the number of points and the degree were

entered. The output consisted of two parts: The first contained:

1) The coefficients of the fitting polynomial

2) An estimate of the error of each coefficient

3) Sum of the square of the deviations

4) The F-ratio

5) The Chi Square goodness of fit statistic

6) Degrees of Freedom of the data, and

7) A list of the data points, the evaluated polynomial at each value of X, and the differences between the two y values.

Part two is a plot of:

1) The data points,

2) the fitted curve, and

3) the normalized error.

The computer program and sample output can be found in Appendix B.

4. <u>Least-Squares Regression Method</u>

The least-squares regression method of data fitting is the most familiar and the most widely used by engineers. It is used by hypothesizing the existence of a polynomial that will best fit the data as in the case of orthogonal polynomials. The degree of the fitting polynomial must be determined by another method or by analysis of the results in order to determine the "best" fit. In the case of an Nth order fit, the fitting polynomial has the form:

$$y = a + bx + cx^2 + dx^3 + --- + nx^N$$

Additional equations are needed to solve for the coefficient a, b, c, d, ..., n. These are generated by multiplying the equation by x, x^2, x^3, etc., then summing the variable on both sides to obtain:

$$\begin{aligned}
\sum y &= Ma + b\sum x + c\sum x^2 + d\sum x^3 + \dots + n\sum x^n \\
\sum xy &= a\sum x + b\sum x^2 + c\sum x^3 + d\sum x^4 + \dots + n\sum x^{n+1} \\
\sum x^2 y &= a\sum x^2 + b\sum x^3 + c\sum x^4 + d\sum x^5 + \dots + n\sum x^{n+2} \\
&\vdots \\
\sum x^n y &= a\sum x^n + b\sum x^{n+1} + c\sum x^{n+2} + d\sum x^{n+3} + \dots + n\sum x^{2n}
\end{aligned} \qquad (1)$$

where

M is the number of points.

In matrix notation Equation (1) can be written:

$$\begin{Bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & \dots & A_{1n} \\ A_{21} & A_{22} & A_{23} & \dots & A_{2n} \\ \vdots & & & & \\ A_{n_1} & A_{n_2} & A_{n_3} & \dots & A_{nn} \end{bmatrix} \begin{Bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{Bmatrix} \qquad (2)$$

or as:

$$\bar{Y} = \bar{A}\ \bar{X} \qquad (3)$$

EXAMPLE 2 Least-squares regression method

As an example, the data of example one will be fitted to a third order polynomial. The data in Table I can be used to generate the data in Table II.

TABLE II

Least-Squares Data

	x	y	x^2	xy	x^3	x^2y	x^4	x^3y	x^5	x^6
	0	5	0	0	0	0	0	0	0	0
	2	21	4	42	8	84	16	168	32	64
	4	133	16	532	64	2128	256	8512	1024	4096
	6	437	36	2622	216	15732	1296	94392	7776	46656
	8	1029	64	8232	512	65856	4096	526848	32768	262144
	10	2005	100	20050	1000	200500	10000	2005000	10000	100000
Σ	30	3630	220	31478	1800	284300	15664	2634920	51600	412960

Substituting the data from Table II into the Equation (1)

$$\begin{aligned} 3630 &= 6a + 30b + 220c + 1800d \\ 31478 &= 30a + 220b + 1800c + 15664d \\ 284300 &= 220a + 1800b + 15664c + 51600d \\ 2634920 &= 1800a + 15664b + 51600c + 412960d \end{aligned} \tag{4}$$

Solving (4) for the coefficients gives

$a = 5$, $b = 0$, $c = 0$, $d = 2$;

therefore the resulting equation is

$$y = 5 + 2x^3$$

which is the exact solution to the data being fitted.

B. CALCULATOR APPLICATIONS

Experimentation with representative data on the IBM 360/67 computer showed that a polynomial of seventh order was required to best-fit a curve smoothly through the data points. This was also verified using the difference method. The least-squares regression method of curve fitting was implemented on the calculator solving for a seventh order polynomial. Various problems were encountered while trying to solve Equation (1) for the unknown coefficients (a,b,c, ..., n) because the A matrix was highly ill-conditioned. Two methods were used to try to solve the equation. The first was a direct approach using triangular factorization with forward and back substitution. The second was an iterative method using a Gauss-Seidel iterative scheme with and without successive over-relaxation (SOR). Examples of these methods can be found in reference 4.

The triangular factorization did not produce acceptable results because of the loss of significant digits. The calculator solution differed so extremely from the computer solution and the plot of the resulting polynomial equation using the calculator solution bore so little resemblance to actual performance plot that other methods had to be tried. Scaling of the input data had no correcting effect on the results using this procedure.

The Gauss-Seidel iterative method occasionally would provide the proper coefficients for the fitting polynomial;

however, this was also an unsatisfactory method as a general solution to the problem. See Figure 2 for the flow chart of the calculator program implementing the Gauss-Seidel iterative method. The initial guess that was used to arrive at the proper solution was extremely critical. An improper guess would lead to instability and the coefficients would diverge from the proper solution. Even when stable and converging, the time required for the calculator to reach a solution was excessive. An iteration required about 20 seconds and, depending on the initial guess, a problem could require over 200 iterations to arrive at a solution which would closely approximate the physical phenomenon. The introduction of a relaxation factor helped; however, like the initial guess, the choice was critical and no way was developed to aid in the choice of either an initial guess or a relaxation factor.

The time problem and the initial guess and relaxation factor problems caused this method to be unacceptable as a general solution to the problem of curve fitting on the Monroe 1880 calculator.

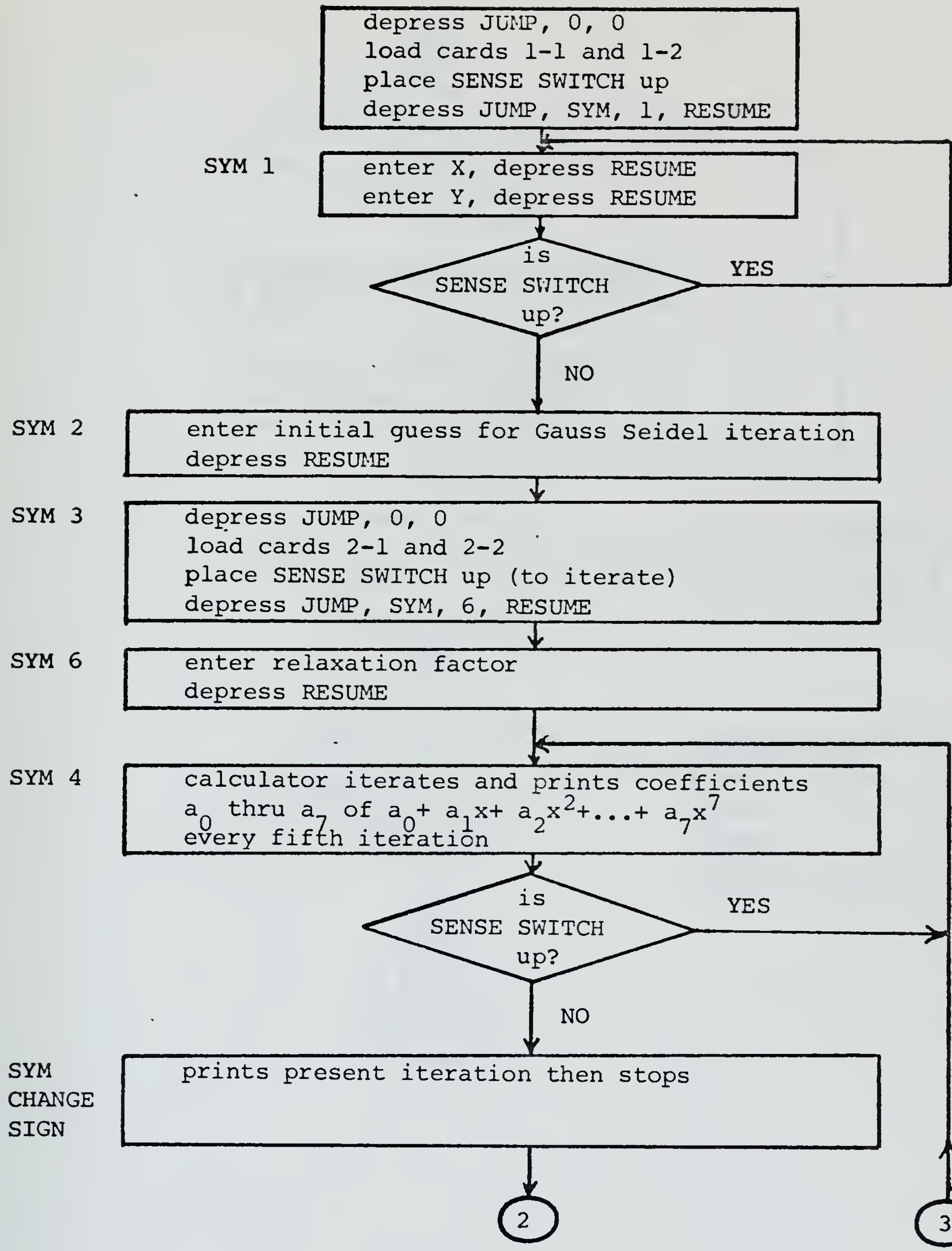


Figure 2. Flow Chart of Calculator Curve Fitting Routine.

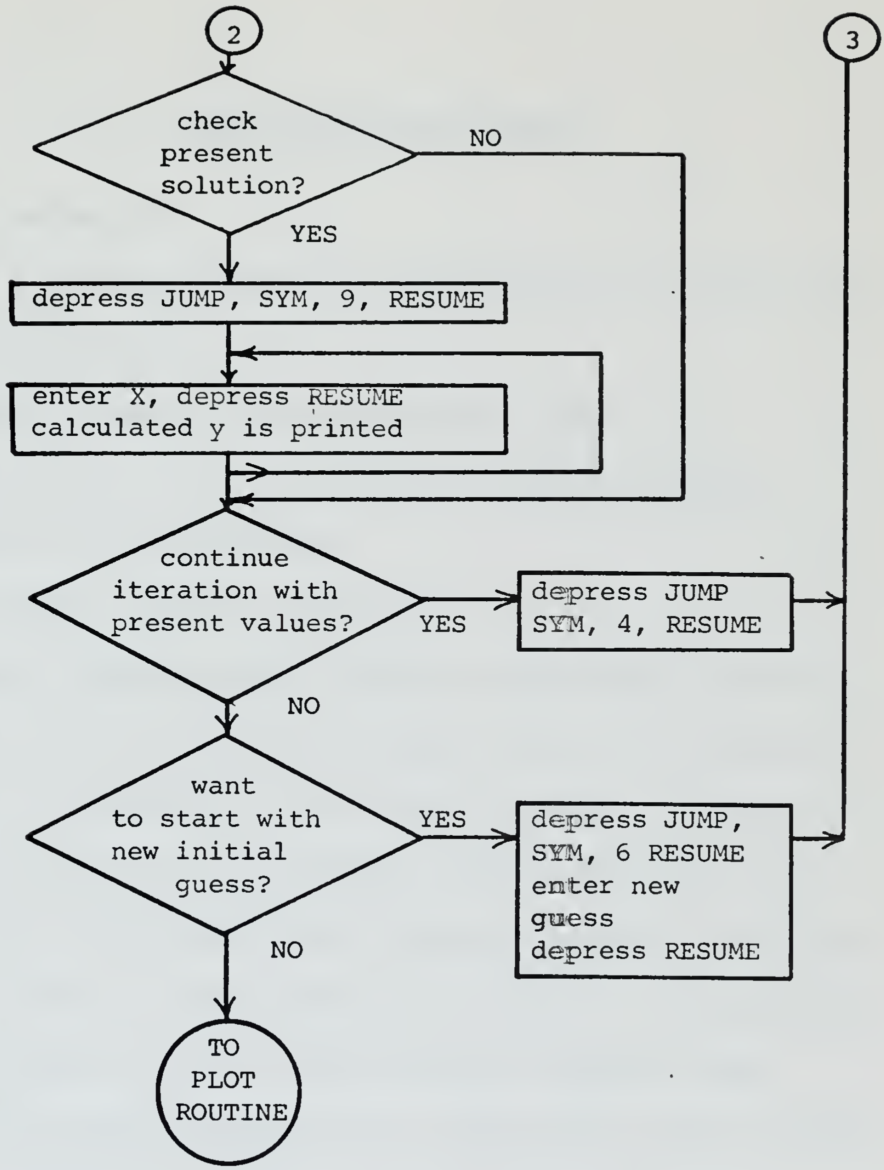


Figure 2. (continued)

IV. DATA REDUCTION SYSTEM

A. SYSTEM COMPONENTS

The data reduction system is composed of the following five major components:

1. Signal selector and conditioner unit
2. A/D converter and calculator interface module
3. Monroe 1880 calculator
4. Monroe PL4 digital plotter
5. Hewlett Packard model 7100B strip chart recorder

The data in the form of an analog signal are played back from all 14 channels of the Pemco magnetic tape recorder. These signals go into the signal selector and conditioner unit from which the operator chooses the path that he wants each of the signals to follow. He can send up to eight signals to the A/D converter module also located in the signal selector and conditioner unit, which in turn feeds digital raw data to the Monroe 1880 calculator for manipulation and plotting, or he can send the raw analog data to the multi-channel strip chart recorder. This flexibility is essential to the operation of an efficient data reduction system.

1. Signal Selector and Conditioner Unit

The signal selector and conditioner module was designed to allow for operator ease in selecting, monitoring,

and routing signals from the tape recorder to the other components of the data reduction system. All channels of the recorder feed into the module through the first 14 terminals of a CO-ORD Switch 10 x 15 pin, 2-deck Matrix Program Board. By selecting the desired output channel, the operator can filter, amplify and/or sum signals using any of nine filter/amplifier circuits and a summing circuit on the 10 outputs of the matrix (see Figure 3). The signal voltage can be monitored at any of 24 points using a built-in Datel model DM 2000AR Digital Panel Meter and a 24 position rotary selector switch. The monitoring points include each of the inputs from the 14 tape recorder channels, each of the outputs from the nine filter/amplifier circuits and the output from the summing circuit.

Signal conditioning was needed to filter out the extraneous high frequency noise from the tape recorder signals which originated from the XR-3 onboard sensors. The noise results principally from vibrations of the testcraft during normal operations; consequently each of the signals has about the same characteristic frequency noise as shown in Figure 4. The filters used are a Butterworth low pass type [Ref. 5] with a cutoff frequency of 0.5 Hz. The electronic circuit diagram and the signal to frequency response curve are shown in Figure 5. The circuit was constructed using a Philbrick Nexus SQ-10a amplifier.

The output from the filter is then amplified using a Burr Brown 3440J amplifier circuit. The one-volt

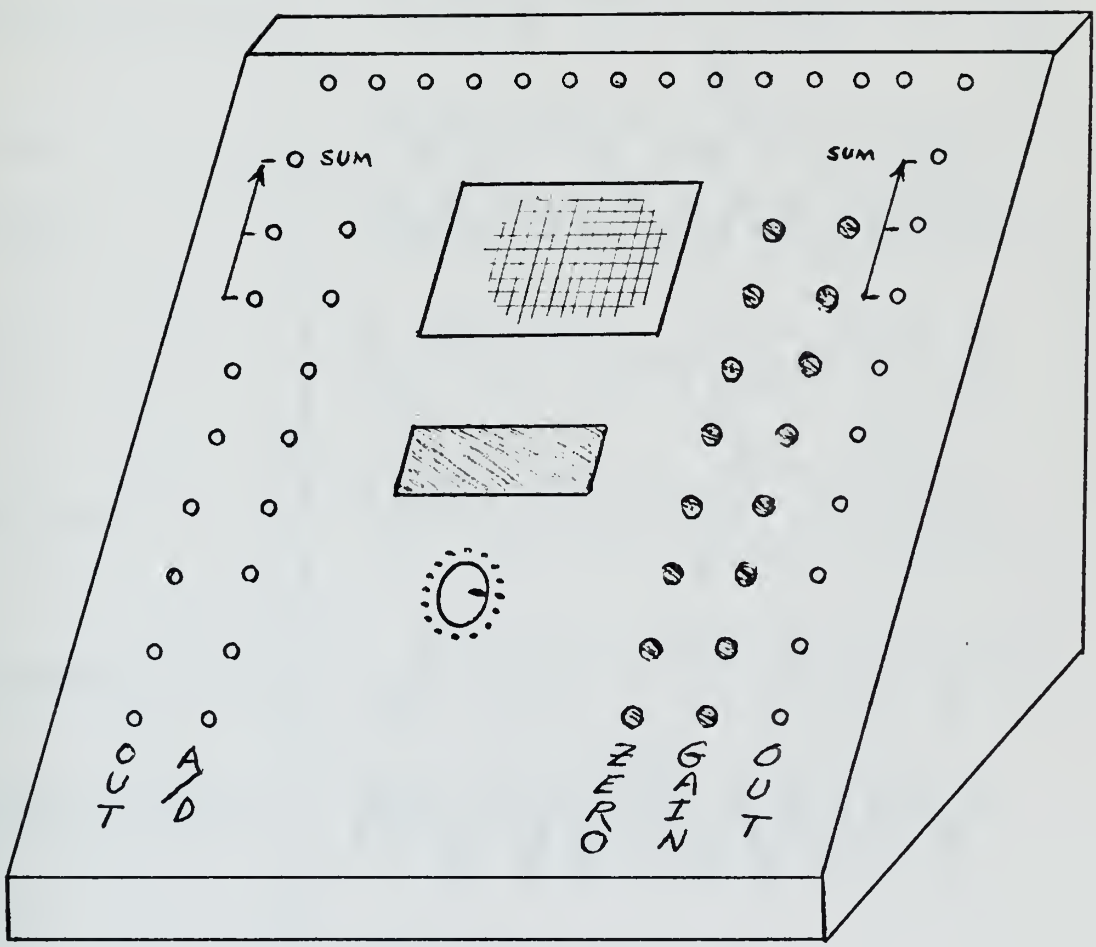
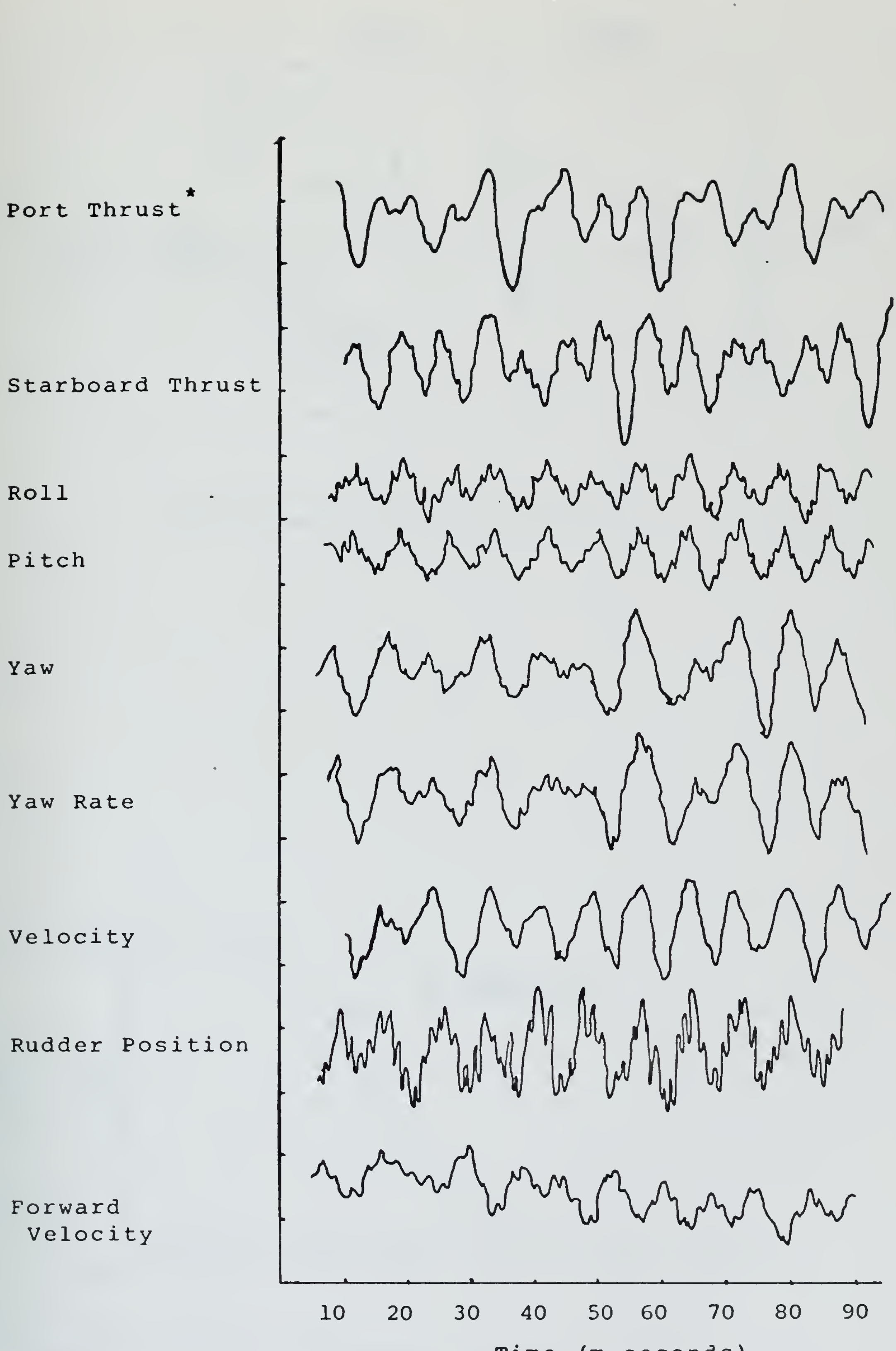
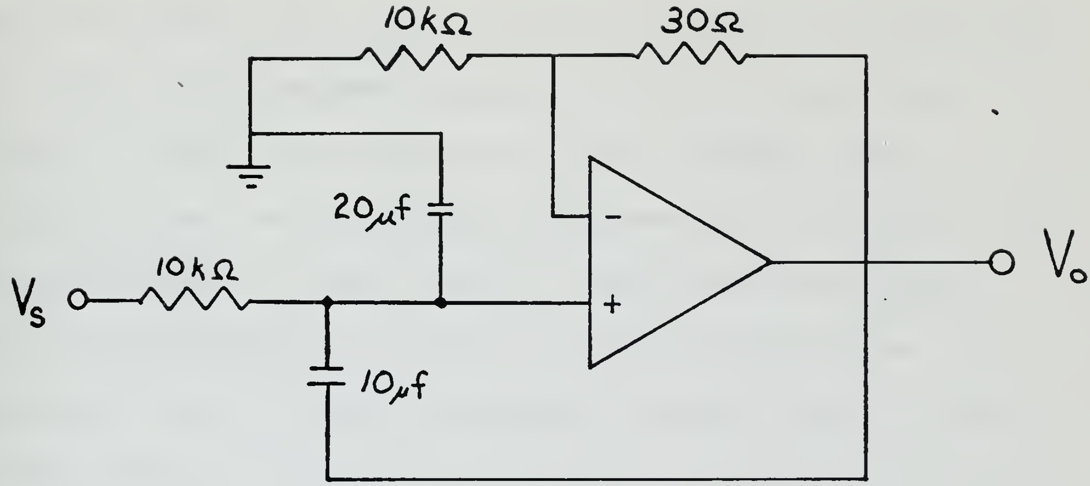


Figure 3. Signal Selector and Conditioner Unit.



*1 cm = 0.1 volt

Figure 4. Sensor Output Signals.



Butterworth Filter Circuit

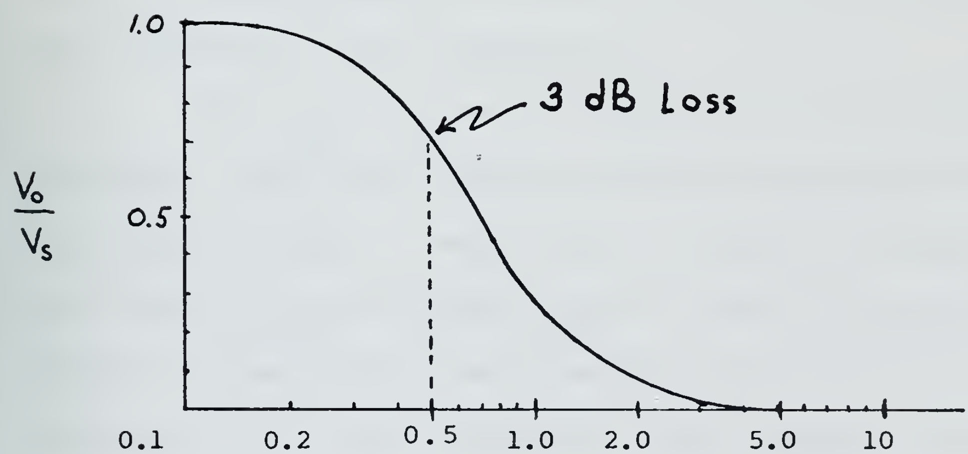
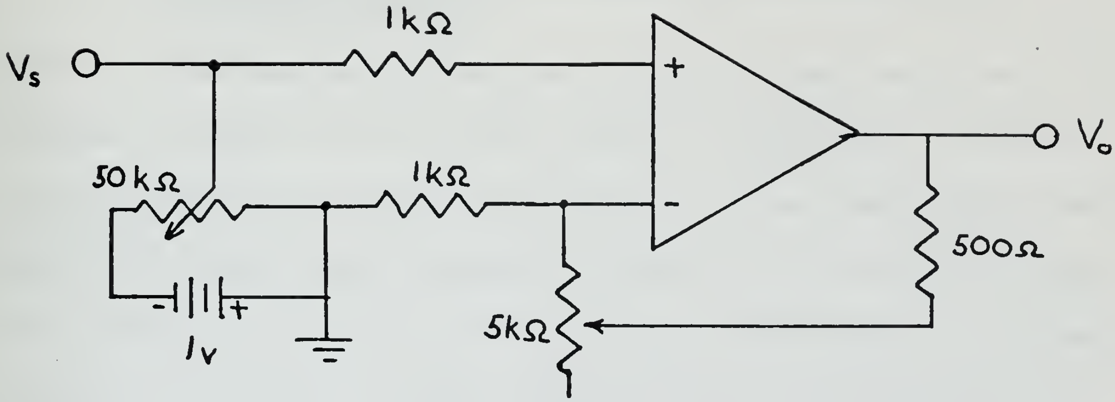


Figure 5. Filter Frequency Response Diagram.

maximum signal from the tape recorder must be amplified to a five-volt maximum signal for use with the A/D converter. This is to ensure the best possible data accuracy when using the A/D converter and calculator interface module. Since each of the sensor/amplifier/ signal conditioner circuits on the testcraft does not maintain a null output for a zero signal input, a zero offset voltage circuit was included. Inasmuch as a full scale input does not always have an exactly 1.0 volt output, a variable gain resistor was placed in the feedback circuit of the amplifier to allow the operator to adjust the gain to have a 5.0 volt output for a full scale input. See Figure 6 for the amplifier circuit.

Two of the filter/amplifier circuits are added then halved through a summing circuit using another Burr Brown 3440J amplifier (see Figure 7). This circuit is used in obtaining total thrust from the individual port and starboard thrust components.

Electrical power is supplied to the Burr Brown and Philbrick amplifiers from a Philbrick/Nexus NPS-300 +15v -15v power supply. Zero offset voltage is supplied by a SRC MOD 3564 power supply and the Datel Digital Panel Meter is powered by a Acopian MOD 10J75 10v dc power supply reduced to 5v dc by a National Semiconductor LM340K-7805 5v dc regulator.

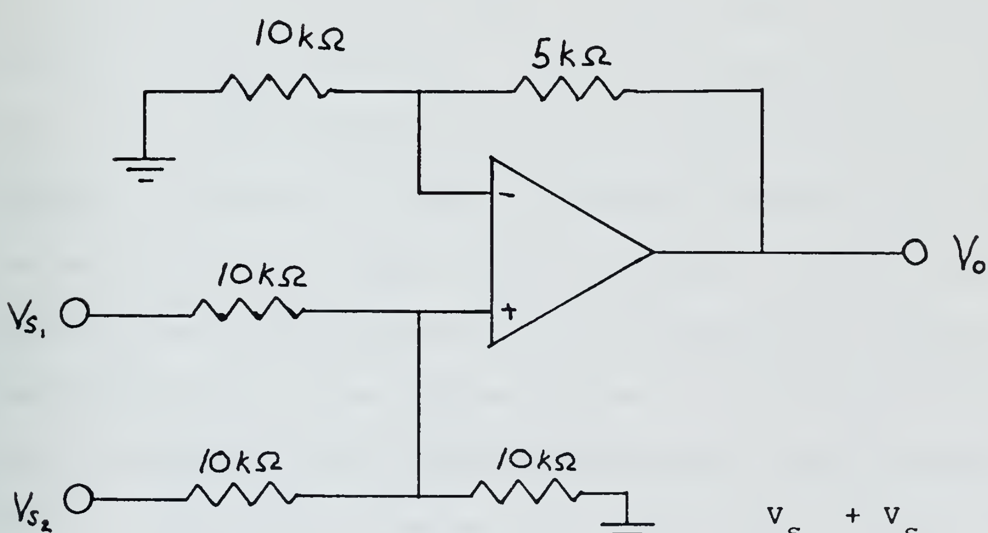


$$0 \leq V_s \leq 1.0$$

$$0 \leq V_0 \leq 5.0$$

$$5(V_s - V_m) = V_0$$

Figure 6. Amplifier Circuit.



$$\frac{V_{s_1} + V_{s_2}}{2} = V_0$$

Figure 7. Summer Circuit.

2. Analog-to Digital Converter and Calculator Interface Module

The analog-to-digital converter and calculator interface module, also called the A/D interface, was purchased from Santa Cruz Engineering in late July 1973. Many modifications were required to permit it to be used with the Monroe calculator and the other components of the data reduction system. These modifications are discussed in Section V.

The A/D interface is selected through connections on the front of the signal selector unit. As many as eight signals can be connected and generally the outputs from the filter/amplifiers are used as the inputs. The calculator program selects which input signal is to be converted by sending a three-digit octal code from the calculator IX register to a TTL/MST 74164, S-bit, serial to parallel converter in the interface. The first three bits are then sent to a RCA COS/MOS analog multiplexer CD4051A for the channel selection. The multiplexer receives the input signals from all eight input channels. It acts as a binary controlled analog switch by selecting which channel is to be outputted to the A/D converter. The A/D converter is then triggered by a TTL/MONOSTABLE 9602 multivibrator.

The A/D converter is a high speed, highly accurate and stable Analogic MP 2410 unit housed in a small 4 inch x 2 inch x 0.4 inch package. It is capable of ten-bit

resolution; however, only eight bits are used for compatibility with the Monroe 1880. It is capable of ranges of -10V to +10V, -5V to +5V and 0V to +5V with output in one's complement, two's complement, offset binary or unipolar binary. The 0V to +5V range using unipolar binary was selected for the data reduction system application. The output from the A/D converter then goes into a TTL/MSI 7491, eight-bit, serial-in serial-out, shift register. Then upon another command from the calculator, the binary signal is transferred from the shift register in the interface to the IX register of the calculator for further processing.

Timing for the interface is provided by the A/D converter for the internal function and by the calculator for data transfer. Power is provided by an Analogic AN3001 modular power supply outputting +15V, -15V and +5V dc. Timing and transfer logic is provided by FAIRCHILD TTL/LSI 7400 Quad two-input NAND GATE and 7410 triple three-input NAND GATE integrated circuits. The logic diagram for the A/D interface can be found in Figure 8.

3. Monroe 1880 Calculator

The Monroe Model 1880 Scientific Programmable Printing Calculator was chosen as the heart of the data reduction system because it combines the features of a simple-to-operate but powerful scientific calculator with

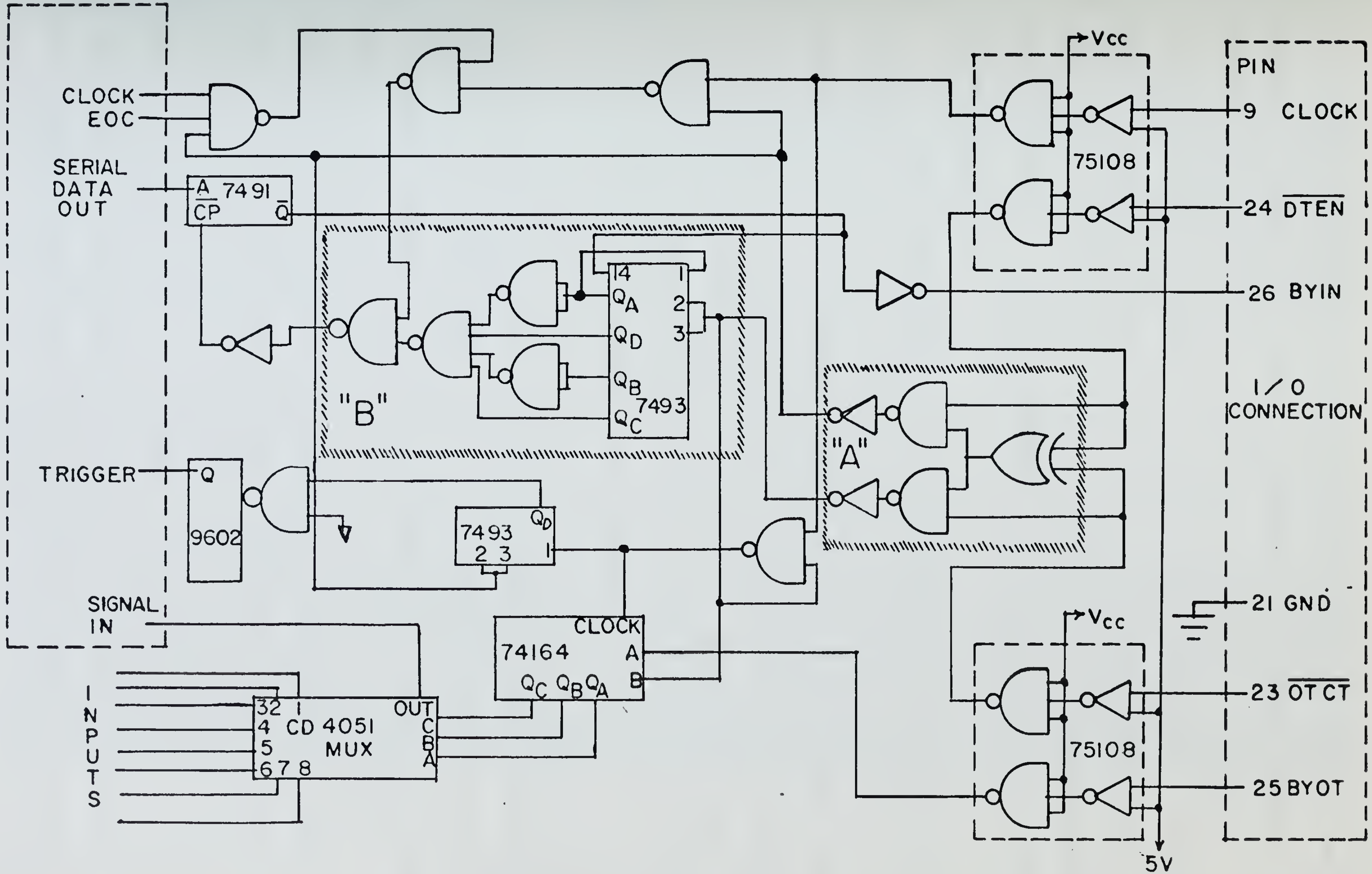


Figure 8. A/D Converter Logic Diagram.

an extensive problem-solving capability. The model being used provides 1024 memory locations for program steps (instructions), 128 main data registers and 10 temporary (scratch pad) registers (both for data storage). The main data registers can be programmed to hold an additional 1024 instruction steps instead of data if a larger program is necessary. Data is stored to an accuracy of 13 significant digits.

The operations of the calculator fall under six general categories: input, output, storage, control, execution and arithmetic computation. All of these operations can be programmed from the keyboard either through individual function keys or through an ENTER CODE key which allows the three-digit octal code of the instruction to be entered into memory (see Figure 9).

A program can also be stored using a magnetic card device which is built into the calculator and enables programs to be both read into the calculator and written on to small magnetic cards each capable of holding 512 program steps. Data from the main data registers can also be written on the cards and conversely data from the cards can be read into the main data registers. Data may be inputted to and outputted from peripheral devices through the IX register and an external 36-pin connector to the peripheral. Programs can be printed on paper tape with the listing containing the instruction number, the three-digit octal code and a symbol to identify the instruction

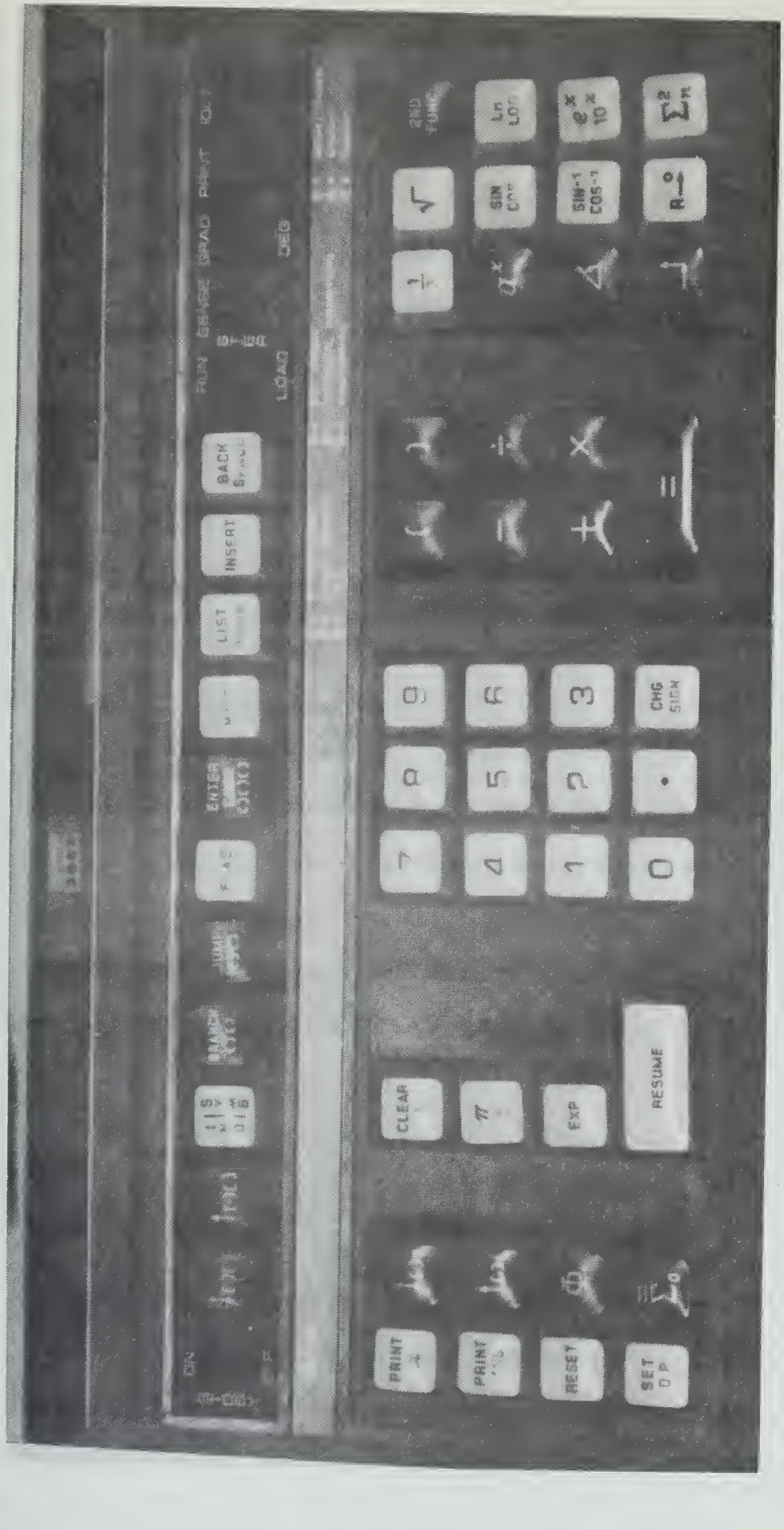


Figure 9. Monroe 1880 Keyboard.

if one is applicable. Program steps can also be inserted, deleted or changed without having to re-enter the entire code. This feature is a tremendous aid in correcting errors in a program.

Control functions include such items as sitting flags, printing data, and setting the position of the decimal point. Program execution is extremely easy with the Monroe 1880. Symbolic program addressing makes it possible to locate a program anywhere in program memory. It permits branching or jumping to any step in a program or to the beginning of a subroutine without having to know the absolute storage location address. Branching returns to the point of the branch after the execution of a reserve; a jump does not. Decision making is made simple with the use of a sense switch, a programmable and a keyboard flag and through the test of a positive, negative or zero value of the contents of the main entry register. Tests of these devices are used to control branches or jumps to other parts of the loaded program.

The arithmetic functions performed by the calculator include such standard keyboard items as add, subtract, multiply, divide, square root and reciprocal. In addition, other keyboard items such as sine/cosine, arc sine/arc cosine, logarithm, statistical summation and factorial are included. Many more functions and controls are available through enterable instruction codes for which keys are not available. These include most of the codes for transferring

information to peripheral devices, program control, register manipulation and other arithmetic functions such as absolute value and arc tangent. A complete set of instruction codes and operating instructions can be found in Refs. 6, 7, and 8.

4. Monroe PL4 Digital Plotter

The Monroe PL4 Digital Plotter is a peripheral device used in association with the Monroe 1800 series calculators. Its purpose is to display graphically data which is passed to it as a result of program execution. The graph origin may be set anywhere on the plotting surface through the use of a calculator scaling routine which also formats the data for transfer to the plotter. The plot subroutine discussed in Section IVB and shown in Appendix D, determines whether the plotter is "busy" (actually plotting) or is ready to receive data. If busy, the program loops until the plotter is ready at which time it passes the pen command and the data for plotting.

Instructions and data are passed to the plotter through the calculator IX register. The four commands which can be issued by the calculator are presented in Table III.

TABLE III

Command Codes for Monroe PL4 Plotter

CODE	COMMAND
160	X data to be passed
161	Y data to be passed
162	PEN UP
163	PEN DOWN

The plotter responds to the PEN UP and PEN DOWN commands only when the REMOTE switch is depressed (see Figure 10). In the manual mode of operation, when the REMOTE switch is not depressed, the pen responds to the commands issued by the operator. These commands are:

WRITE (pen down)
PEN UP
MOMENTARY (spring loaded up, pen down while depressed)

and are issued through switches on the plotter control panel. After the X data or the Y data instruction has been passed to the plotter, the calculator transfers the data from memory to the IX register for transmission to the plotter. See Table IV for further plotter characteristics.

TABLE IV

Monroe PL4 Digital Plotter Characteristics

Plotting Area	10 inches x 10 inches
Ballpoint Pen Width	0.009 inch
Resolution (X and Y axis)	1000 counts
Repeatability	0.1 percent of full scale
Accuracy	
Static	0.2 percent of full scale
Dynamic	3 inch retrace will overlay trace within 0.035 inches between center lines
Dimensions	
Width	17 inches
Length	21 inches
Height	7.75 inches
Weight	48 pounds

5. <u>Hewlett Packard Model 7100B Strip Chart Recorder</u>

The Hewlett Packard 7100B Strip Chart Recorder (Figure 11) has two independent pen drives and input modules which can accommodate analog input up to 100 V. The chart

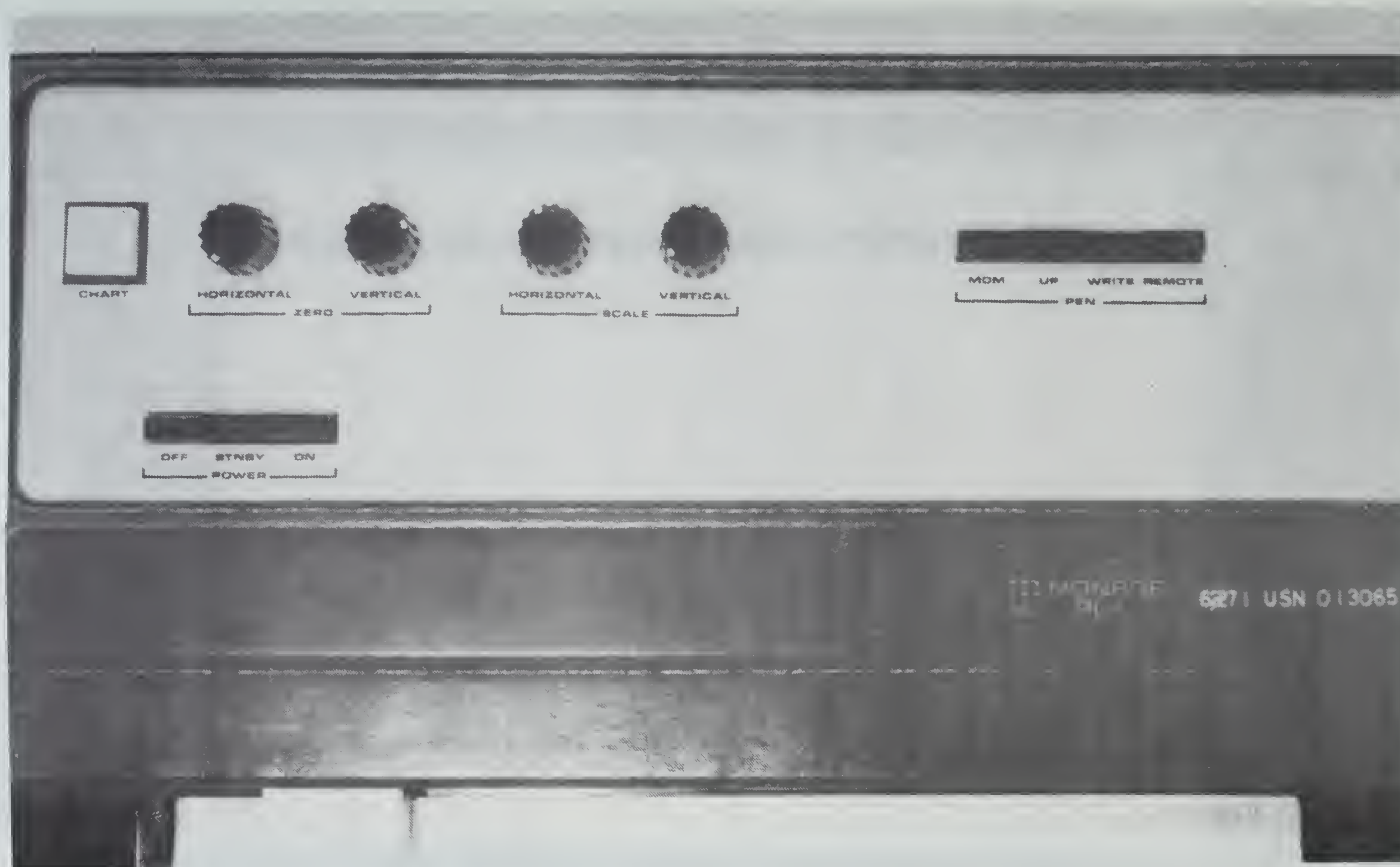


Figure 10   Monroe PL4 Plotter Control Panel.

Figure 11. Hewlett Packard Model 7100B Strip Chart Recorder.

transport system has 12 selectable speeds and uses a standard chart roll with a 10-inch writing width. The pens move horizontally across the top of the chart each capable of a full 10-inch travel. Specifications are listed in Table V.

TABLE V

HP 7100B Strip Chart Recorder Specifications

Response time	0.5 seconds full scale
Chart Speeds	1.2 in/h; 0.1, 0.2, 0.5, 2 in/min 0.1,0.2,0.5,1,2 in/s
Input Range	Variable from 1mV to 100 V
Input Resistance	1 megohm
Dimensions	
Width	16-3/4 in
Height	8-11/16 in
Depth	7-1/4 in
Weight	20 pounds

B. CALCULATOR PROGRAMMING AND DATA TRANSFER

1. <u>Memory</u>

The calculator memory is divided into four pages. Each page is then further divided into columns with each containing 32 registers, numbered in octal from 00-37. The first page, Page 0, is user's programmable memory and contains four columns. Each column can hold 256 instruction codes giving a total capacity of 1024 instruction storage locations. The second page (Page 1) is data register memory also containing four columns for a total of 128 data registers used for main data storage. The third and fourth pages contain 16 columns each consisting

of working storage and read-only memory (ROM). The ROM contains the permanently stored programs that perform the keyboard functions. The last two columns of the fourth page (Page 3) containing the working storage, and all of the first two pages are made up of random access memory (RAM) which can be written into and read from. Information stored in Pages 0, 2 and 3 is treated as a program instruction. A single instruction (or program step) occupies one byte (8 bits). Eight single instructions can be stored one item per register. The format is shown in Figure 12. Each byte contains two Binary Coded Decimal (BCD) digits except the sign position.

The calculator contains three programmable control registers, the Entry (E) Register, the data Address Register (DC) and the Index (IX) Register. The entry register is the primary data register. Inputs from the keyboard are put in the E register. Output data are formatted there, then transferred to a print buffer and at the end of any calculation the E register contains the result. The entry register is formatted the same as a data register with the addition of an overflow and an underflow digit. The address (DC) register contains 14 bits and contains the address of a specific location in memory. It is used with various macro instructions for defining the address to be used for data or code manipulation.

The index (IX) register is a general purpose 8 bit register. It is divided into two 4 bit parts, IH (index

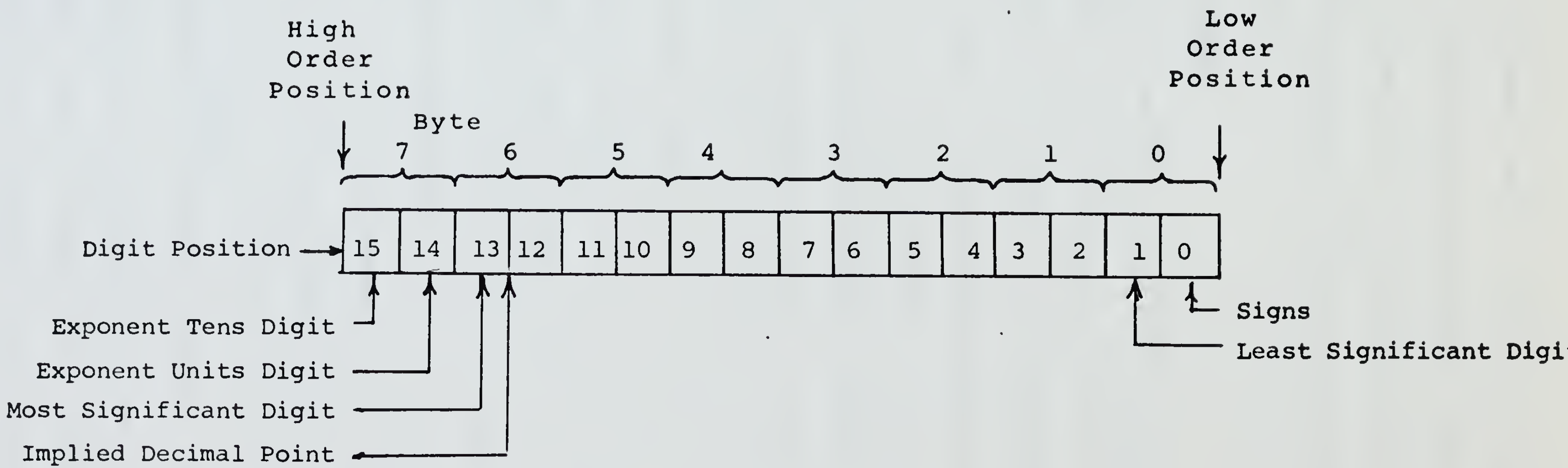


Figure 12. Data Register Format.

high) and IL (index low). The IX is used as a buffer register for the input and output (I/O) of information to peripherals through the I/O bus. All information transfer between peripherals and the calculator occurs through the IX register.

2. <u>I/O Lines</u>

The data output line from the calculator to the I/O bus is BYOT (byte, output, pin 25 on the connector). Data from the IX register is transmitted on this line on the falling edges of the clock pulses from T8-T15 time (see Figure 13) during an OPCB (output control byte, code 314) command and during an OPDB (output data byte, code 315) command. The OPCB command is used to transmit selection and control data of peripheral devices while the OPDB command will send numeric data over the output line.

The data input line into the calculator from peripherals is BYIN (byte, input, pin 26). Data is transmitted from the peripheral when an IPDB (input data byte, code 305) command is executed.

The calculator transmits two signals to the peripheral device to allow the device to determine which type of command, control or data I/O, the calculator is executing. The signal $\overline{\text{OTCT}}$ (output control time, pine 23) remains at a logical 0 from T8-T15 time and $\overline{\text{DTEN}}$ (data transfer enable, pin 24) goes to a logical 1 when the calculator executes an OPCB (output control byte). If an OPDB (output data byte) or an IPDB (input data byte)

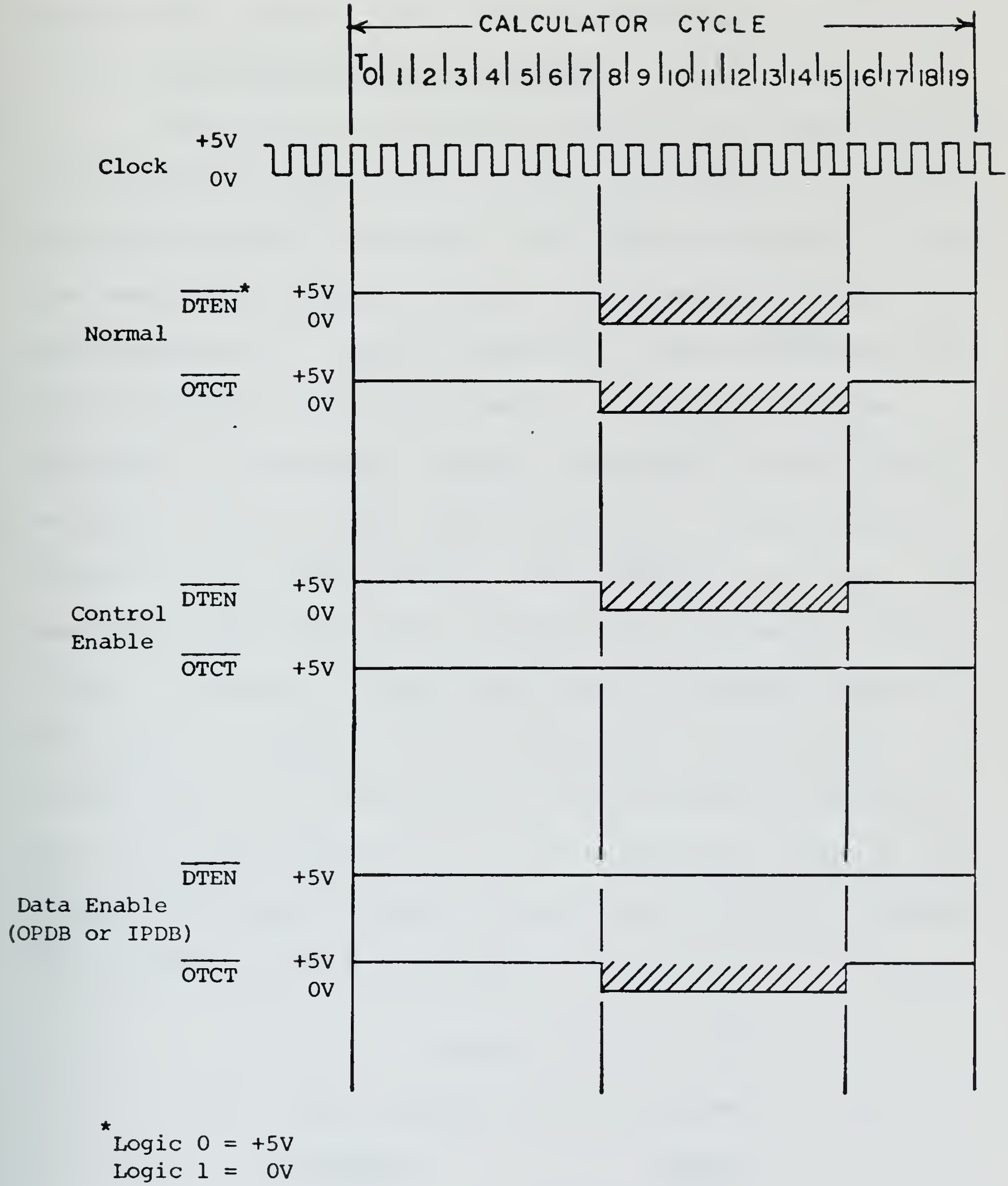


Figure 13. Data Transfer Signals.

command is executed, $\overline{\text{DTEN}}$ will remain at a logical 0 from T8-T15 time and OTCT will go to logical 1.

3. A/D Interface Data Transfer

The operator determines which of the input signals to the A/D interface are converted and sent to the calculator by entering the proper code (shown in Table VI) into the computer program at decimal step numbers 130 and 135 (see Appendix D). During execution, the calculator loads the IX register with the code for the channel of the dependent (y) variable (decimal steps 129 and 130) then outputs a control byte (OPCB code 314) to set the multiplexer to the proper channel then convert the signal. The next step (IPDB code 305) transfers the converted signal to the IX register. These data are then stored and the code for the channel of the independent (X) variable is loaded into the IX register and the process is repeated. After the two signals are in the calculator, they are converted from binary to BCD, scaled, and printed if desired, then passed to the X-Y plotter.

TABLE VI

A/D Interface Channel Codes

CHANNEL	CODE
1	340
2	140
3	240
4	040
5	300
6	100
7	200
8	000

4. Digital Plotter Data Transfer

The digital plotter has a slightly more complicated data transfer scheme because of its slower response time. The first 42 instructions (000-041) comprise the plotter subroutine and are used to control data input to this device. The main program branches into the subroutine at the decimal steps 000, 004, 008 and 012. When a value is to be passed, rather than a pen command, the value is stored in the E register and is register 27 of the last column. For example, when transfer of Y data to the plotter is desired, the program branches to step 004. The program in Appendix E can be followed.

The code for Y data (161) is loaded into the IX register than the subroutine jumps to octal step 16. The contents of the IX register is then copied into another register. The IX retains a copy, then the calculator outputs a control byte (OPCB). The program then checks the lower 4-bits of the IX register (IL) to be not equal to 0. Since the IL register contains a 1, a jump to octal address 25 (decimal address 21) is performed. A RINS (read input status) is executed and the plotter returned a 1 to the H8 (left most) bit of IX if busy and the calculator repeats until not busy or the plotter returns a 0. The IX register is then re-loaded with the initial Y code (161) destroyed by the last step then tested for a pen command. This test fails so the sign digit of the E register is passed to IL then transferred to the plotter

(OPDB). The address register (DC) is loaded with the address of register 27 of the last column by the instructions 273 and 277. This register contains the number in the E register; however, data transfer is easier from this location. IX is loaded with the contents of the address pointed to by DC which is the two exponent digits. These are transferred to the IX register then DC is decremented by one step (two digits) and the transfer is repeated. The plotter receives the exponent, the four most significant digits of the number and the sign of both. The plotter pen then moves to the point.

V. A/D INTERFACE PROBLEMS AND SOLUTIONS

Initially the A/D interface had four features which inhibited it from being used with the rest of the data reduction system. The two major problems were (1) the logic for data transfer and (2) the loss of the most significant bit from the A/D converter. The two less serious and more easily corrected faults were (3) the voltage conversion range of the A/D converter and (4) the conversion of binary to decimal numbers.

A. LOGIC FOR DATA TRANSFER

The initial design of the A/D interface assumed both $\overline{\text{DTEN}}$ and $\overline{\text{OCTC}}$ to remain logical 0's throughout the entire normal time cycle instead of both going high from T8-T15 time. It also assumed that $\overline{\text{DTEN}}$ would go high for data transfer and $\overline{\text{OCTC}}$ would go high for control output instead of missing the one time pulse train. Consequently the A/D interface could not interpret the calculator requests. In order to correct this design flow, the circuit labeled "A" in Figure 8 was incorporated. The two signals, $\overline{\text{OCTC}}$ and $\overline{\text{DTEN}}$, are compared with an exclusive OR. If the two are different, the output of the exclusive OR is high. The original $\overline{\text{OCTC}}$ and $\overline{\text{DTEN}}$ are then "AND"ed to determine if either signal has been missed indicating the command has been executed. This addition changes the $\overline{\text{OCTC}}$ and $\overline{\text{DTEN}}$ signals to that expected by the rest of the interface.

B. LOSS OF MOST SIGNIFICANT BIT

After correcting the first problem, analog signals could be converted to binary and transferred to the calculator. However, only seven of the eight bits were being passed to the IX register. It was determined that the most significant bit was not transferring into the interface 8-bit shift register. It was left "floating" at the output of the A/D converter then lost when the first transfer pulse was being issued by the calculator. The circuit labeled "B" in Figure 8 was added to remove the first transfer pulse during an IPDB and allows the entire 8 bits to be transferred into the IX register. It uses a TTL/MST 7493 4-Bit Binary Counter and four "NOR" gates configured to remove the T9 time pulse from the clock to the 8-bit shift register during the transfer of data from the A/D converter.

C. VOLTAGE CONVERSION RANGE

The Analog-to-Digital converter was wired at delivery to convert signals in a range of -5V to +4.96. Since all the signals from the XR-3 are in a range of OV to +1V, this larger range reduced the significance which could be achieved by the converter. Using eight bits coded binary, the full scale range of the converter can be broken down into 2^8 or 256 parts. The smallest range capable of the A/D converter is OV to +4.98V so in order to realize the greatest possible accuracy two changes had to be made.

First the A/D converter range was changed to 0V to +4.98V and second the signal condition and amplifier unit was developed and built to enable the tape recorder signals to be amplified to a 5V maximum signal. By making these two changes the full capability of the A/D converter is utilized.

D. BINARY TO DECIMAL CONVERSION

The calculator data format is binary coded decimal (BCD) characters. Four bits are used to represent a decimal digit and a number is made up of by combining BCD characters. The A/D interface transfers a number to the IX register in binary representation. In order to be used by the calculator, this number has to be converted to BCD format. This is done by calculator software by a series of jump instruction beginning at decimal step 48 of the data reduction program. IX register bit L1 (right most) is the most significant bit. If this bit is high, 128 is added to scratch pad register 0. If bit L2 is on, 64 is added, and so on such that if all bits are on, scratch pad register 0 totals 255. After checking all bits of the IX register, the scratch pad register is divided by 256. The result of this is to leave in the register the decimal equivalent of the 0v to 1V analog signal converted by the A/C interface.

EXAMPLES OF CONVERSION:

VOLTAGE IN	A/D OUTPUT	CALCULATOR CONVERSION
0.25V	0100 0000	$2^6/2^8 = 64/256 = 0.25$
0.50V	1000 0000	$2^7/2^8 = 128/256 = 0.50$
0.6875V	1011 0000	$(2^7+2^5+2^4)/2^8 = (128+32+16)/256$ $= 0.6875$
1.0V	1111 1111	$(2^7+2^6+2^5+\ldots+2^0)/2^8 = \frac{255}{256} = .998$

VI. CONCLUSIONS

The automatic data reduction system has been shown to provide a quick and accurate means of evaluating performance data from the XR-3 testcraft. It has been tested with both simulated and actual tape recorder signals and proven to be accurate to greater than 1% of full scale. Raw data can be efficiently transferred into a finished graphical presentation and a printed record of the data, converted to physical units, can be easily obtained. The system is superb for its simplicity and ease of operation. A job that used to require days can now be completed in a matter of hours.

The signal selector and interface unit provides the capability of zeroing and calibrating the signals from the tape recorder. It removes the signal noise resulting from vibration and can pass the data to a strip chart recorder, or it can internally digitize the information for transfer to the calculator for further processing and/or plotting.

The provision for expansion has been designed into the system both through the modification of hardware and software. The signal selector and conditioner unit has room for internal modification and additions to satisfy requirements which may arise in the future, and the calculator has over six columns of unused memory which may be used to increase the data reduction program to fulfill future needs in this area.

APPENDIX A

XR-3 DATA ACQUISITION SYSTEM

The data acquisition system presently installed in the XR-3 testcraft is comprised of various sensors feeding through amplifiers and/or signal conditioners in order to prepare the signal for recording on a Pemco Model 120-B magnetic tape recorder (see Figure 14).

A. SENSOR SYSTEM

Sensors are presently installed to measure the following parameters:

1. Port thrust
2. Starboard thrust
3. Forward seal pressure
4. Aft seal pressure
5. Plenum chamber pressure[1]
6. Testcraft velocity
7. Water emersion height
8. Pitch angle
9. Pitch rate
10. Roll angle
11. Roll rate
12. Yaw angle
13. Yaw rate
14. Lateral acceleration
15. Longitudinal acceleration

Thrust is measured by a Revere USPI-150A (30) balance bridge transducer. One is installed on each engine mount so as to measure the fore or aft thrust of each engine.

---

[1]Two pressure probes are installed in the plenum, one forward and one aft. Only one can be used at a time.

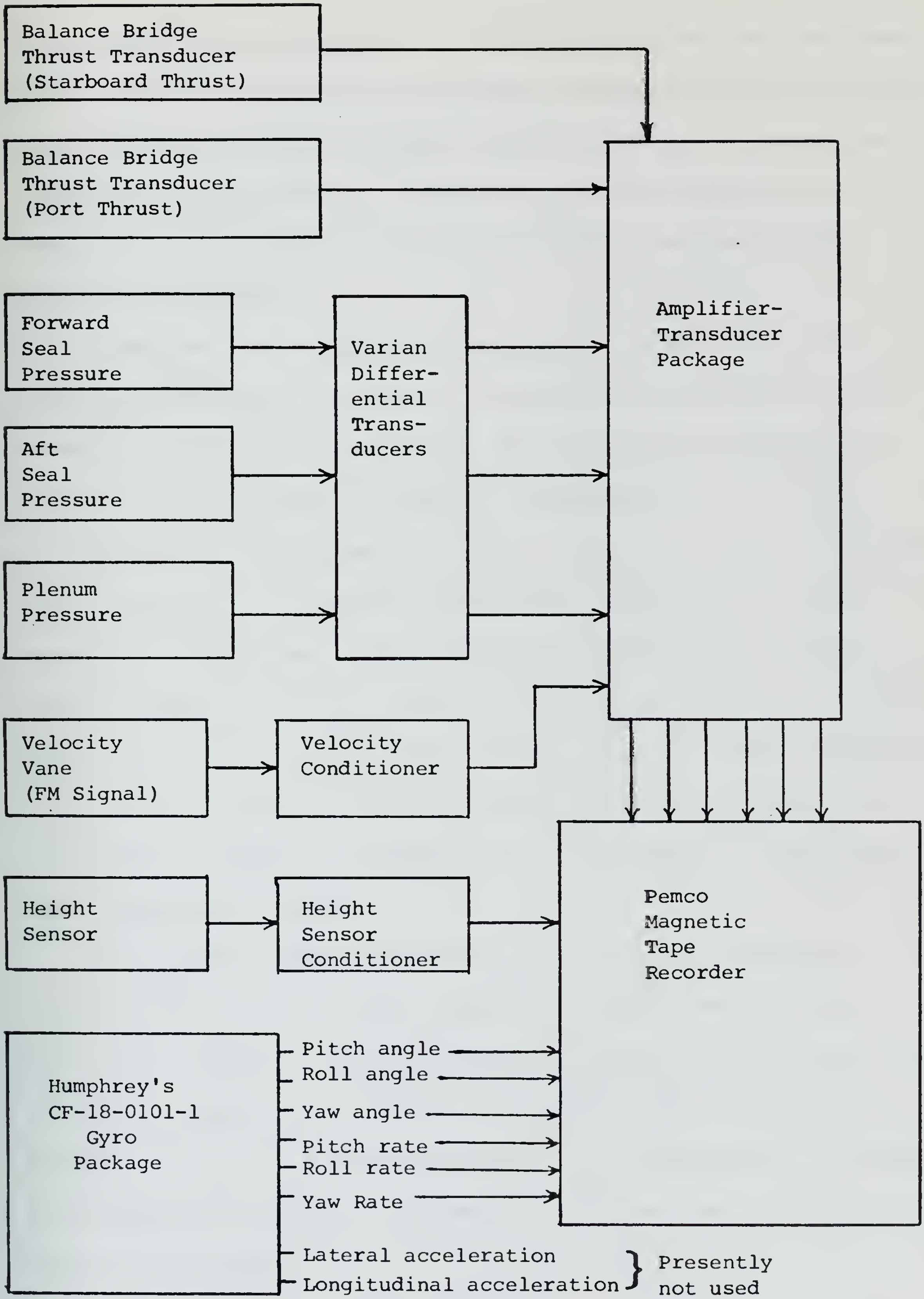


Figure 14. Data Acquisition System.

The signal from the bridge is amplified by one of ten Grant Model DCAB-3 amplifiers to a range between 0.0 and 1.0 volts corresponding to 0.0 to 500.0 pounds thrust. Calibration signals of 0.0, 0.5 and 1.0 volts are also available to each of the amplifiers which are found in an amplifier-transducer package.

Pressures are measured by pneumatic probes connected to Varian differential pressure transducers accurate to $\pm$ 1%. Pressure signals are amplified and calibrated similar to thrust so as to have 60 lbf/ft^2 correspond to 1.0 volt.

Velocity is measured by a vane enclosed in a water probe which transmits a frequency modulated signal to a Potter velocity conditioner. The velocity conditioner converts the FM signal to a dc signal from 0.0 to 5.0 volts corresponding to 0.0 to 40.0 knots. This signal is then reduced to 0.0 to 1.0 volt by another amplifier in the amplifier-transducer package. Calibration is performed by operation over a measured course.

A Humphreys Model CF18-0101-1 gyro package measures pitch angle and rate, roll angle and rate, yaw angle and rate, and lateral and longitudinal acceleration. Presently all modes except acceleration are being used. Outputs are between 0.0 and 1.0 volt and range is $\pm$ 20 degrees in pitch, $\pm$ 15 degrees in roll, $\pm$ 180 degrees in yaw, and $\pm$ 30 degrees /sec in all rates.

Height is measured by a Western Marine Electronics Model LM4001 height sensor with the transducer located forward of the bow. Range is selectable ± 2 feet or ± 3 feet, both with an accurate of 1%. The ± 2 feet range is presently in use. The signal from the sensor is converted to 0.0 to 1.0 volt dc by a height sensor conditioner.

B. DATA RECORDER

The conditioned signals from the various on board sensors are sent to and recorded on a Pemco Model 120-B magnetic reel-to-reel tape recorder. The recorder has 14 channels for electronic data plus an edge track for audio recording. The unit can be controlled from either the pilot's instrument panel or from the unit itself, located just aft of the pilot's cockpit. The recorder is portable weighing only 100 pounds and measuring 9-1/2" x 17-3/4" x 26-1/2". It is easily installed and removed from the testcraft.

The recorder can operate at tape speeds of 60, 30, 15, 7-1/2, 3-3/4, and 1-7/8 inches per second and uses standard or precision NAB 10-1/2 inch reels and 1.0 or 1.5 mil one-inch tape.

APPENDIX B

IBM 360/67 COMPUTER DATA FITTING

A fortran program was developed for use on the IBM 360/67 computer to aid in calculator software design. The program consists of five basic parts as follows:

Part 1: Initialization and control

Part 2: Data point plotting

Part 3: Determination of fitting polynomial coefficients and error criteria

Part 4: Error normalization and plotting

Part 5: Evaluation and plotting of final curve

The plotting subroutine, DRAW, is used to provide the final graphical output. It is available as a callable subroutine and utilizes a Calcomp Plotter.

The subroutine LSF provides the fitting polynomial coefficients and all the written output. It is a modification of the subroutine LSQF2 and the procedures and algorithm for the method can be found in Ref. 2.

```
C      ***************************************************************
C
C      PURPOSE
C       TC FIT A CURVE TO A SET OF DATA PCINTS USING THE
C       LEAST-SQUARES METHCD WITH ORTHOGCNAL POLYNCMIALS
C
C      INPUT
C       1. N=NUMBER CF PCINTS TO FIT (FCRMAT I2)
C       2. THE DATA UP TO 60 PCINTS (FORMAT 2F10.3)
C       3. THE CEGREE OF FIT (1 THRU 20 FCRMAT I2)
C       4. TITLE FOR THE CUTPUT AND PLOT (FIRST 48 COLUMNS CF
C          2 CARDS; FIRST CARD: TITLE, SECOND: NAME)
C
C      ***************************************************************
C
C
       DIMENSICN FPT(200),XPT(200),X1(30,2),F1(30,2),X2(60)
      X ,DEL(60)
       REAL*8 TITLE(10),LABEL/'          '/,TITLE2(12),DMAX
       REAL*8 X(60),F(60),Y(60),WI(60),DELY(60),B(21),SB(21)
       DATA WI/60*1.0D0/,LABE1/' ERR'/,LABE2/'      '/
       CATA TITLE/10*'        '/,TITLE2/12*'        '/
       LAST=0
       K=5
       L=6
       READ (K,91)N
C
C    TC READ IN THE DATA POINTS
C
       READ(K,93)(X(I),F(I),I=1,N)
       DC 7 I=1,30
       X1(I,1)=X(I)
       F1(I,1)=F(I)
       X1(I,2)=X(I+30)
       F1(I,2)=F(I+30)
       X2(I)=X(I)
       X2(I+30)=X(I+30)
 7     CCNTINUE
 20    READ (K,91)KM
       IF (KM.GT.25)GO TC 30
       KM=-KM
 8     I=1
       READ (K,200)TITLE2
       DC 12 J=1,6
       TITLE(J)=TITLE2(J)
 12    CCNTINUE
 9     NPTS=30
       IF (N.LT.NPTS)NPTS=N
       IF ((I.EQ.2).AND.(N.LT.60))NPTS=N-30
       CALL DRAW(NPTS,X1(1,I),F1(1,I),I,1,LABEL,TITLE2,
      X.2,.2,1,0,2,2,5,6,0,LAST)
       IF (N.LT.30)I=2
       I=I+1
       IF (I.LT.3)GO TO 9
C
C   TC PLCT THE ERROR
       CALL LSF(N,KM,X,F,WI,Y,DELY,B,SB,TITLE)
       DMAX=0.0
       DC 36 I=1,N
       IF (DABS(DELY(I)).GT.DMAX)DMAX=DABS(DELY(I))
 36    CCNTINUE
       DC 37 I=1,N
       DEL (I)=DELY(I)/DMAX/10.
 37    CCNTINUE
       CALL DRAW(N,X2,DEL,2,0,LABE1,TITLE2,
      X.2,.2,1,0,2,2,5,6,0,LAST)
```

FIGURE 15. FORTRAN CURVE FITTING ROUTINE

```
C     TC PLCT THE FUNCTION CURVE
C
        KM=-KM
        XNT=(X(N)-X(1))/200.0
        XPT(1)=X(1)
        FPT(1)=POLY(KM,XPT,B,1)
        DC 10 I=2,200
        XPT(I)=XPT(I-1)+XNT
        FPT(I)=PCLY(KM,XPT,B,I)
 10     CCNTINUE
        CALL DRAW(200,XPT,FPT,3,0,LABE2,TITLE2,EXSC,
      X.2,.2,1,0,2,2,5,6,0,LAST)
        GC TO 20
 90     FORMAT (1F10.4)
 91     FCRMAT (I2)
 92     FORMAT (I3)
 93     FCRMAT(2F10.3)
 98     FCRMAT (' DMAX=',F10.4)
 200    FORMAT (6A8)
 30     STOP
        ENC


C     ..................................................
C
C      FUNCTICN POLY
C      PURPOSE
C       TC EVALUATE THE PCLYNCMIAL USING NESTING
C      INPUTS
C               1. N=THE DEGREE OF THE PCLYNCMIAL FIT
C               2. X= THE ARRAY OF POINTS AT WHICH TO EVAL THE
C                     POLYNCMIAL
C               3. B= THE ARRAY OF CCEFFICIENTS
C               4. I=THE SUBSCRIPT OF THE X
C
        FUNCTION POLY(N,X,B,I)
        REAL*8 B(21), DX, C
        DIMENSION X(200)
        DX=X(I)
        C=B(N+1)
        DC 5 K=1,N
        J=N-K+1
        C=B(J)+C*DX
 5      CONTINUE
        PCLY=C
        RETURN
        END
```

FIGURE 15.(CONTINUEC)

```
C     ...............................................................
C
C      SUBROUTINE LSF
C      PURPOSE
C       TO SOLVE FOR THE LEAST-SQUARES POLYNOMIAL
C      INPUTS
C       1. M=NUMBER OF DATA POINTS(LESS THEN 100)
C       2. KM=THE DEGREE OF FIT
C       3. X=ARRAY OF X VALVES
C       4. F2=ARRAY OF Y VALUES
C       5. WI=ARRAY OF WEIGHS FOR THE DATA POINTS
C       6. Y=OUTPUT ARRAY OF THE FIT DATA
C       7. DELY=OUTPUT ARRAY OF ERROR (F2-Y)
C       8. B=OUTPUT ARRAR OF THE COEFFICIENTS OF THE
C          POLYNOMIAL FIT
C       9. SB=OUTPUT ARRAY OF THE ESTIMATED ERROR IN B
C       10. TITLE=HEADING FOR THE OUTPUT
C         NOTE: X,F2,WI,Y,DELY,B,SB,AND TITLE ARE REAL*8
C
C
       SUBROUTINE LSF(M,KM,X,F2,WI,Y,DELY,B,SB,TITLE)
       REAL*8 X(1),F2(1),WI(1),Y(1),DELY(1),B(1),SB(1),
     X    FM,FMR,PXF,PXP,XPXP,XPXPM,PPXPP,XP,ALPHA,BETA,PPXF,
     X F(100),P(100),PM(100),S(100),ST(100),A(21,21),T(100),
     X SIG2,SIG3,SUMEV2,FLEV,F2BAR,SQ,FMF,FMKF,AM,CHI,W(100)
     X ,TITLE(1),FBAR,XBAR
       DO 1 I=1,21
       DO 1 J=1,21
     1 A(I,J)=0.0
       A(1,1)=1.0
       A(2,2)=1.0
       PBAR=0.0
       SUMEV2=0.D0
       FM=0.0
       DO 5 I=1,M
       IF(WI(I).EQ.0.0) WI(I)=1.0D0
     5 FM=FM+WI(I)
       FMR=1.0D0/FM
       FBAR=0.0
       XBAR=0.0
       DO 10 I=1,M
       W(I)=WI(I)*FMR
       PM(I)=DSQRT(W(I))
       F(I)=F2(I)*PM(I)
       SUMEV2=SUMEV2+F2(I)*WI(I)
       FBAR=FBAR+F(I)*PM(I)
    10 XBAR=XBAR+X(I)*W(I)
       T(1)=FBAR
       A(2,1)=-XBAR
       PXF=0.0
       PXP=0.0
       F2BAR=SUMEV2/FM
       SUMEV2=0.D0
       DO 20 I=1,M
       SUMEV2=SUMEV2+ W(I)*(F2(I)-F2BAR)**2
       P(I)=(X(I)-XBAR)*PM(I)
       PXF=PXF+P(I)*F(I)
    20 PXP=PXP+P(I)*P(I)
       MFM = FM + 1.D-5
       T(2)=PXF/PXP
       PMXPM=1.0D0
       S(1)=PMXPM
       KMP=IABS(KM)+1
       B(1)=T(1)*A(1,1)+T(2)*A(2,1)
       B(2)=T(2)*A(2,2)
       DO 1000 K=2,KMP
       CHI=0.
       KM1=K-1
       KM2=K-2
```

FIGURE 15.(CONTINUED)

```
      IF(KM2.EQ.0) GO TO 200
      XPXP=0.0
      XPXPM=0.0
      B(K)=0.0
      DO 50 J=1,M
      XP=X(J)*P(J)
      XPXP=XPXP+XP*P(J)
   50 XPXPM=XPXPM+XP*PM(J)
      ALPHA=XPXP/PXP
      BETA=XPXPM/PMXPM
      PPXF=0.0
      PPXPP=0.0
      DO 100 I=1,M
      PT=P(I)
      P(I)=(X(I)-ALPHA)*P(I)-BETA*PM(I)
      PPXF=PPXF+P(I)*F(I)
      PPXPP=PPXPP+P(I)*P(I)
  100 PM(I)=PT
      T(K)=PPXF/PPXPP
      PMXPM=PXP
      PXP=PPXPP
      A(K,1)=-ALPHA*A(KM1,1)-BETA*A(KM2,1)
      A(K,KM1)=A(KM1,KM2)-A(KM1,KM1)*ALPHA
      A(K,K)=1.0
      IF(K.LE.3) GO TO 150
      DO 120 I=2,KM2
  120 A(K,I)=A(KM1,I-1)-ALPHA*A(KM1,I)-BETA*A(KM2,I)
  150 DO 160 I=1,K
  160 B(I)=B(I)+T(K)*A(K,I)
  200 SIG3=0.0
      DO 220 I=1,M
      SQ=B(K)
      KKQ=K-1
      DO 230 IQ=1,KKQ
      KMIQ=K-IQ
  230 SQ=X(I)*SQ+B(KMIQ)
      Y(I)=SQ
      DELY(I)=Y(I)-F2(I)
  220 SIG3=SIG3+ W(I)*DELY(I)**2
      FMKF=MFM-K
      SIG2=SIG3*FM/FMKF
      FLEV=(SUMEV2-SIG3)/SIG2
      SUMEV2=SIG3
      S(K)=PXP
      DO 240 I=1,K
  240 ST(I)=SIG2/S(I)
      DO 300 I=1,K
      SB(I)=0.0
      DO 250 J=1,K
  250 SB(I)=SB(I)+ST(J)*A(J,I)**2
  300 SB(I)=DSQRT(SB(I))
      IF(KM.GT.0)GO TO 301
      CONTINUE
      IF(K.LT.KMP)GO TO 1000
  301 CONTINUE
      WRITE(6,9510)  (TITLE(I),I=1,10)
 9510 FORMAT(1H1,////,30X,10A8,//,25X,'COEFFICIENTS OF THE',
     X' POWER SERIES EXPANSION',//,26X,'Y(X)=B(1)+B(2)*X',
     X'+B(3)*X**2+B(4)*X**3+...'//)
      WRITE(6,9520) ((I,B(I)),I=1,K)
 9520 FORMAT((15X,3('  B(',I2,')=',1PD12.5)))
      WRITE(6,9530)
 9530 FORMAT(//,25X,'ESTIMATES OF ERROR FOR THE',
     X' COEFFICIENTS',//)
      WRITE(6,9540)((I,SB(I)),I=1,K)
 9540 FORMAT((15X,3('  ERRB(',I2,')=',1PD9.3)))
      WRITE(6,9570) SIG3,FLEV
 9570 FORMAT(//,21X,'SUM SQ DEV =',1PD10.3,11X,'F-RATIO ='
     X,1PD10.3)
```

FIGURE 15. (CONTINUED)

```
      DO 25 IRI=1,M
   25 CHI=CHI+DELY(IRI)*DELY(IRI)*WI(IRI)/DABS(Y(IRI))
      WRITE(6,1111)CHI,FMKF
 1111 FORMAT('0',20X,'CHISQ=',1PD11.3,15X,'DEG OF FREEDOM ='
     X,0PF4.0)
      WRITE(6,9550)
 9550 FORMAT(//,34X,'----DATA-----      FIT     DIFFERENCE',/
     2         30X,'I     X(I)     Y(I)    YY(I)    YY(I)-Y(I)',//)
      WRITE(6,9560)((I,X(I),F2(I),Y(I),DELY(I)),I=1,M)
 9560 FORMAT(15X,I16,3F8.3,F11.3)
 1000 CONTINUE
      WRITE(6,9580)
 9580 FORMAT('1')
      RETURN
      END
```

FIGURE 15.(CONTINUED)

VELOCITY VS THRUST 5TH ORDER FIT

COEFFICIENTS OF THE POWER SERIES EXPANSION

Y(X)=B(1)+B(2)*X+B(3)*X**2+B(4)*X**3+...

B(1)= 2.57974D-03 B(2)= 1.10134D 01 B(3)=-9.19896D 01
B(4)= 3.29749D 02 B(5)=-5.34132D 02 B(6)= 3.26051D 02

ESTIMATES OF ERROR FOR THE COEFFICIENTS

ERRB(1)=8.726D-02 ERRB(2)=4.646D 00 ERRB(3)=6.004D 01
ERRB(4)=2.816D 02 ERRB(5)=5.556D 02 ERRB(6)=3.913D 02

SUM SQ DEV = 3.051D-04 F-RATIO = 6.944D-01

CHISQ= 1.271D-02 DEG OF FREEDOM = 9.

I	DATA X(I)	DATA Y(I)	FIT YY(I)	DIFFERENCE YY(I)-Y(I)
1	0.0	0.0	0.003	0.003
2	0.110	0.502	0.467	-0.035
3	0.150	0.406	0.452	0.046
4	0.203	0.402	0.411	0.009
5	0.250	0.411	0.391	-0.020
6	0.275	0.403	0.390	-0.013
7	0.298	0.398	0.396	-0.002
8	0.322	0.411	0.407	-0.004
9	0.351	0.421	0.424	0.003
10	0.372	0.434	0.439	0.005
11	0.398	0.443	0.457	0.014
12	0.427	0.472	0.477	0.005
13	0.458	0.501	0.499	-0.002
14	0.502	0.554	0.539	-0.015
15	0.550	0.622	0.628	0.006

Figure 16. Velocity vs Thrust Output.

VELCCITY VS THRUST 6TH ORDER FIT

COEFFICIENTS OF THE POWER SERIES EXPANSION

Y(X)=B(1)+B(2)*X+B(3)*X**2+B(4)*X**3+...

```
B( 1)= 6.15428D-04   B( 2)= 1.70049D 01   B( 3)=-2.03641D 02
B( 4)= 1.C9230D C3   B( 5)=-2.96328D 03   B( 6)= 3.98231C 03
B( 7)=-2.09999D C3   B(
```

ESTIMATES OF ERROR FOR THE CCEFFICIENTS

```
ERRB( 1)=5.957D-C2   ERRB( 2)=7.585D 00   ERRB( 3)=1.348C 02
ERRB( 4)=8.979D C2   ERRB( 5)=2.820D 03   ERRB( 6)=4.214D 03
ERRB( 7)=2.415C 03   ERRB(
```

SUM SQ,DEV = 1.262D-04 F-RATIO = 7.56CC-01

CHISQ= 4.978D-03 DEG CF FREEDOM = 8.

I	----DATA---- X(I)	Y(I)	FIT YY(I)	DIFFERENCE YY(I)-Y(I)
1	0.0	0.0	0.001	0.001
2	0.110	0.502	0.488	-0.014
3	0.150	0.406	0.434	0.028
4	0.203	0.402	0.392	-0.010
5	0.250	0.411	0.392	-0.019
6	0.275	0.403	0.401	-0.002
7	0.298	0.398	0.409	0.011
8	0.322	0.411	0.418	0.007
9	0.351	0.421	0.427	0.006
10	0.372	0.434	0.433	-0.001
11	0.398	0.443	0.444	0.001
12	0.427	0.472	0.463	-0.009
13	0.458	0.501	0.496	-0.005
14	0.502	0.554	0.564	0.010
15	0.550	0.622	0.619	-0.003

Figure 16. (Continued)

VELOCITY VS THRUST 7TH ORDER FIT

COEFFICIENTS OF THE POWER SERIES EXPANSION

Y(X)=B(1)+B(2)*X+B(3)*X**2+B(4)*X**3+...

```
B( 1)= 1.13850D-04   B( 2)= 2.74653D 01   B( 3)=-4.45397D 02
B( 4)= 3.25500D 03   B( 5)=-1.26547D 04   B( 6)= 2.71388D 04
B( 7)=-3.02829D 04   B( 8)= 1.37332D 04   B(
```

ESTIMATES OF ERROR FOR THE COEFFICIENTS

```
ERRB( 1)=3.674D-02   ERRB( 2)=1.178D 01   ERRB( 3)=2.634D 02
ERRB( 4)=2.304D 03   ERRB( 5)=1.017D 04   ERRB( 6)=2.408D 04
ERRB( 7)=2.918D 04   ERRB( 8)=1.420D 04   ERRB(
```

SUM SQ DEV = 4.200D-05   F-RATIO = 9.356D-01

CHISQ= 1.610D-03   DEG OF FREEDOM = 7.

I	----DATA---- X(I)	Y(I)	FIT YY(I)	DIFFERENCE YY(I)-Y(I)
1	0.0	0.0	0.000	0.000
2	0.110	0.502	0.498	-0.004
3	0.150	0.406	0.417	0.011
4	0.203	0.402	0.392	-0.010
5	0.250	0.411	0.404	-0.007
6	0.275	0.403	0.408	0.005
7	0.298	0.398	0.410	0.012
8	0.322	0.411	0.412	0.001
9	0.351	0.421	0.417	-0.004
10	0.372	0.434	0.425	-0.009
11	0.398	0.443	0.444	0.001
12	0.427	0.472	0.473	0.001
13	0.458	0.501	0.508	0.007
14	0.502	0.554	0.550	-0.004
15	0.550	0.622	0.623	0.001

Figure 16. (Continued)

VELOCITY VS THRUST 8TH ORDER FIT

COEFFICIENTS OF THE POWER SERIES EXPANSION

Y(X)=B(1)+B(2)*X+B(3)*X**2+B(4)*X**3+...

```
B( 1)= 8.75039D-06   B( 2)= 4.58088D 01   B( 3)=-9.48749D 02
B( 4)= 8.81746D 03   B( 5)=-4.49607D 04   B( 6)= 1.34341D 05
B( 7)=-2.34717D 05   B( 8)= 2.22223D 05   B( 9)=-8.80914D 04
```

ESTIMATES OF ERROR FOR THE COEFFICIENTS

```
ERRB( 1)=1.973D-02   ERRB( 2)=1.778D 01   ERRB( 3)=4.773D 02
ERRB( 4)=5.188D 03   ERRB( 5)=2.976D 04   ERRB( 6)=9.795D 04
ERRB( 7)=1.858D 05   ERRB( 8)=1.890D 05   ERRB( 9)=7.979D 04
```

SUM SQ DEV = 1.038D-05   F-RATIO = 1.219D 00

CHISQ= 3.735D-04   DEG OF FREEDOM = 6.

I	----DATA---- X(I)	Y(I)	FIT YY(I)	DIFFERENCE YY(I)-Y(I)
1	0.0	0.0	0.000	0.000
2	0.110	0.502	0.502	-0.000
3	0.150	0.406	0.407	0.001
4	0.203	0.402	0.401	-0.001
5	0.250	0.411	0.408	-0.003
6	0.275	0.403	0.406	0.003
7	0.298	0.398	0.404	0.006
8	0.322	0.411	0.407	-0.004
9	0.351	0.421	0.418	-0.003
10	0.372	0.434	0.431	-0.003
11	0.398	0.443	0.450	0.007
12	0.427	0.472	0.472	0.000
13	0.458	0.501	0.498	-0.003
14	0.502	0.554	0.555	0.001
15	0.550	0.622	0.622	-0.000

Figure 16. (Continued)

VELOCITY VS THRUST 9TH ORDER FIT

COEFFICIENTS OF THE POWER SERIES EXPANSION

Y(X)=B(1)+B(2)*X+B(3)*X**2+B(4)*X**3+...

```
B( 1)= 9.61578D-08   B( 2)= 6.40666D 01   B( 3)=-1.52764D 03
B( 4)= 1.64407D 04   B( 5)=-9.96499D 04   B( 6)= 3.69151D 05
B( 7)=-8.55135D 05   B( 8)= 1.21107D 06   B( 9)=-9.60214D 05
B(10)= 3.26833D 05   B(
```

ESTIMATES OF ERROR FOR THE COEFFICIENTS

```
ERRB( 1)=1.807D-02   ERRB( 2)=5.087D 01   ERRB( 3)=1.590D 03
ERRB( 4)=2.068D 04   ERRB( 5)=1.469D 05   ERRB( 6)=6.263D 05
ERRB( 7)=1.647D 06   ERRB( 8)=2.616D 06   ERRB( 9)=2.303D 06
ERRB(10)=8.628D 05   ERRB(
```

SUM SQ DEV = 7.255D-06 F-RATIO = 1.435D-01

CHISQ= 2.610D-04 DEG OF FREEDOM = 5.

I	----DATA----- X(I)	Y(I)	FIT YY(I)	DIFFERENCE YY(I)-Y(I)
1	0.0	0.0	0.000	0.000
2	0.110	0.502	0.502	0.000
3	0.150	0.406	0.405	-0.001
4	0.203	0.402	0.404	0.002
5	0.250	0.411	0.406	-0.005
6	0.275	0.403	0.404	0.001
7	0.298	0.398	0.404	0.006
8	0.322	0.411	0.408	-0.003
9	0.351	0.421	0.420	-0.001
10	0.372	0.434	0.431	-0.003
11	0.398	0.443	0.448	0.005
12	0.427	0.472	0.470	-0.002
13	0.458	0.501	0.501	-0.000
14	0.502	0.554	0.554	0.000
15	0.550	0.622	0.622	-0.000

Figure 16. (Continued)

VELOCITY VS THRUST 10TH ORDER FIT

COEFFICIENTS OF THE POWER SERIES EXPANSION

Y(X)=B(1)+B(2)*X+B(3)*X**2+B(4)*X**3+...

```
B( 1)=-1.45516D-07  B( 2)= 7.79242D 00  B( 3)= 4.69988D 02
B( 4)=-1.37137D 04  B( 5)= 1.55206D 05  B( 6)=-9.64241D 05
B( 7)= 3.63738D 06  B( 8)=-8.56450D 06  B( 9)= 1.23215D 07
B(10)=-9.92136D 06  B(11)= 3.42870D 06  B(
```

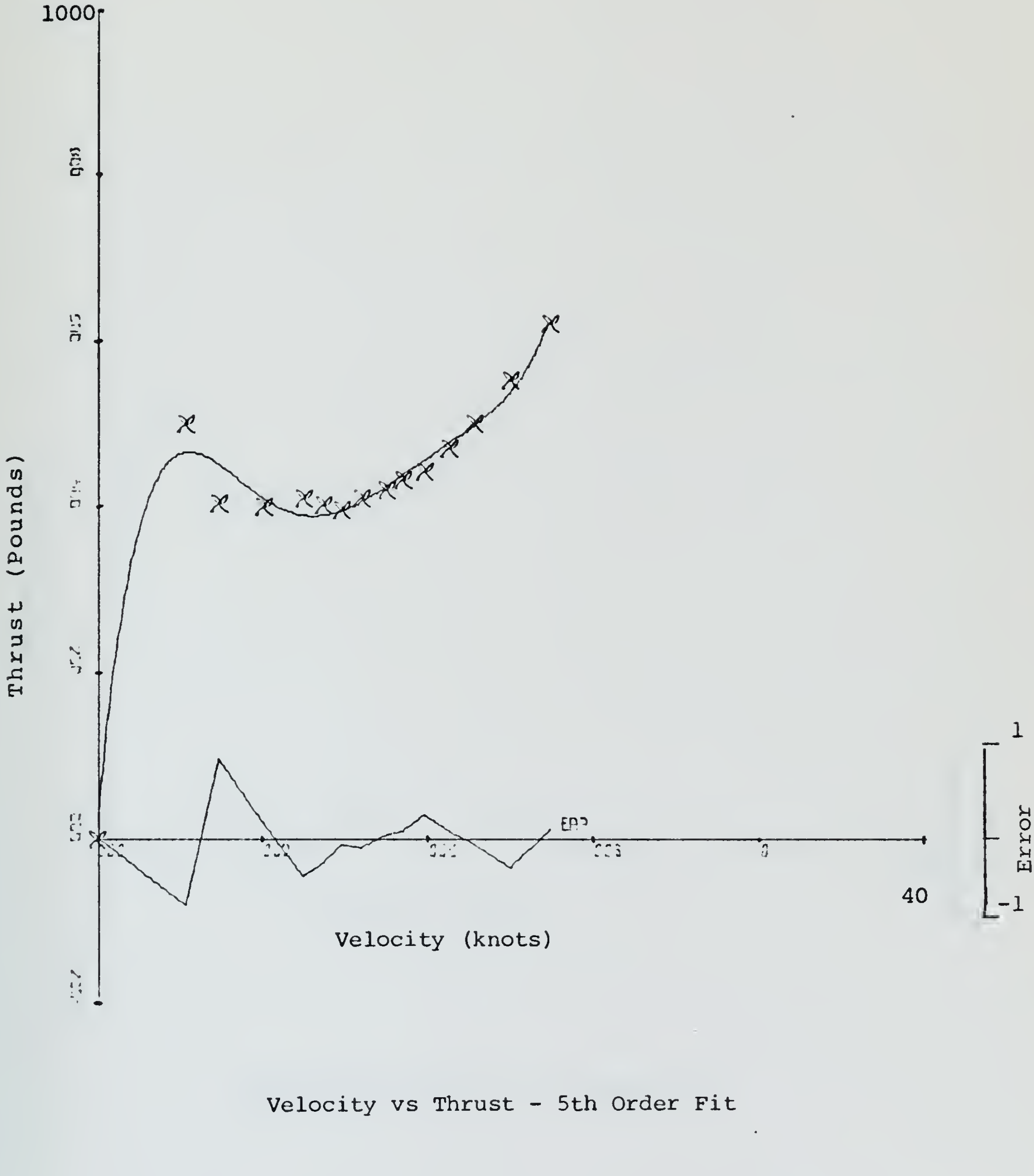
ESTIMATES OF ERROR FOR THE COEFFICIENTS

```
ERRB( 1)=1.777D-02  ERRB( 2)=2.077D 02  ERRB( 3)=7.323D 03
ERRB( 4)=1.099D 05  ERRB( 5)=9.241D 05  ERRB( 6)=4.815D 06
ERRB( 7)=1.617D 07  ERRB( 8)=3.511D 07  ERRB( 9)=4.762D 07
ERRB(10)=3.671D 07  ERRB(11)=1.228D 07  ERRB(
```

```
SUM SQ DEV = 5.615D-06          F-RATIO = 7.784D-02
CHISQ=  1.961D-04               DEG OF FREEDOM =  4.
```

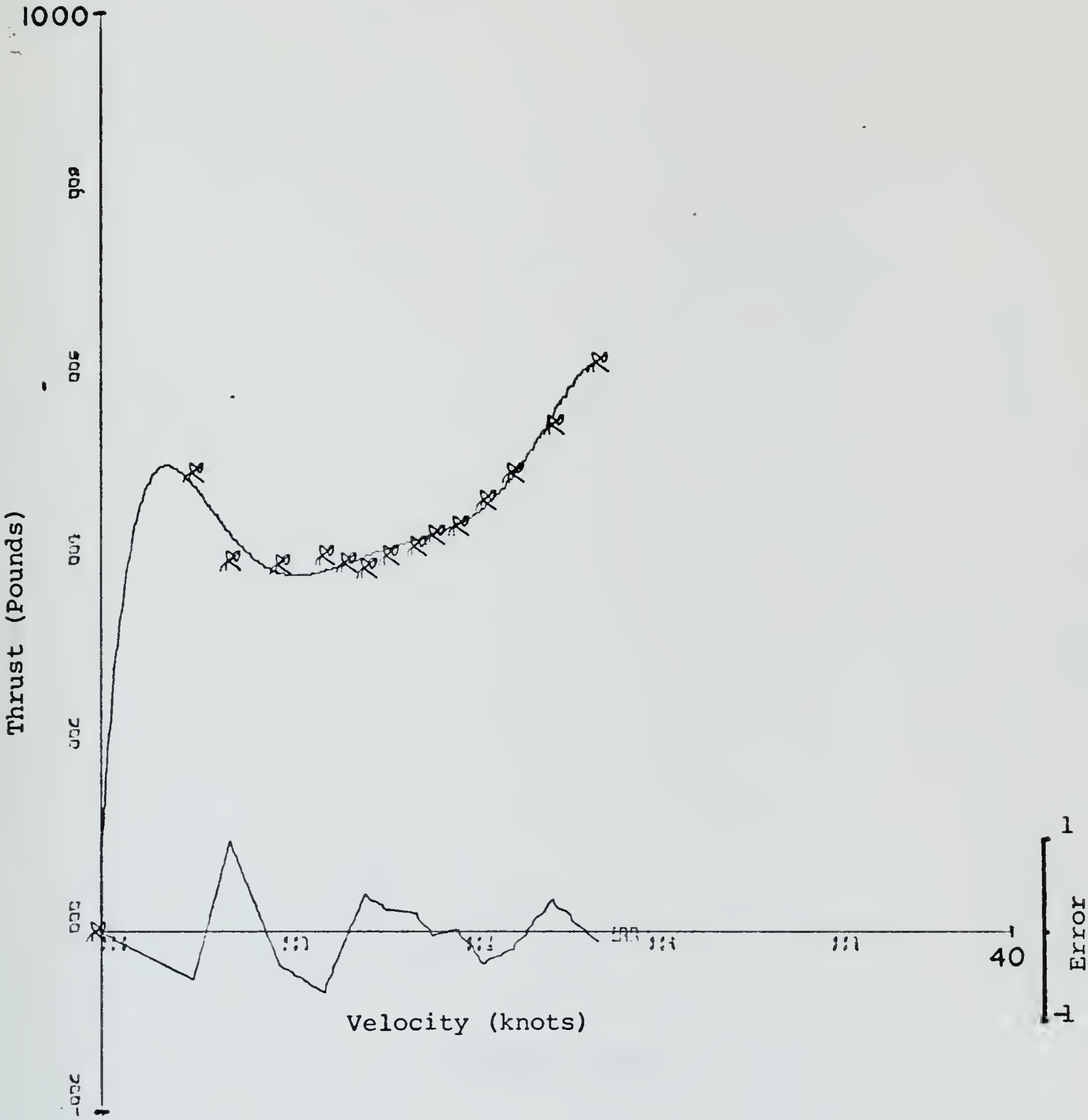
	----DATA-----		FIT	DIFFERENCE
I	X(I)	Y(I)	YY(I)	YY(I)-Y(I)
1	0.0	0.0	-0.000	-0.000
2	0.110	0.502	0.502	0.000
3	0.150	0.406	0.406	-0.000
4	0.203	0.402	0.403	0.001
5	0.250	0.411	0.408	-0.003
6	0.275	0.403	0.404	0.001
7	0.298	0.398	0.402	0.004
8	0.322	0.411	0.407	-0.004
9	0.351	0.421	0.421	-0.000
10	0.372	0.434	0.433	-0.001
11	0.398	0.443	0.448	0.005
12	0.427	0.472	0.468	-0.004
13	0.458	0.501	0.503	0.002
14	0.502	0.554	0.554	-0.000
15	0.550	0.622	0.622	0.000

Figure 16. (Continued)



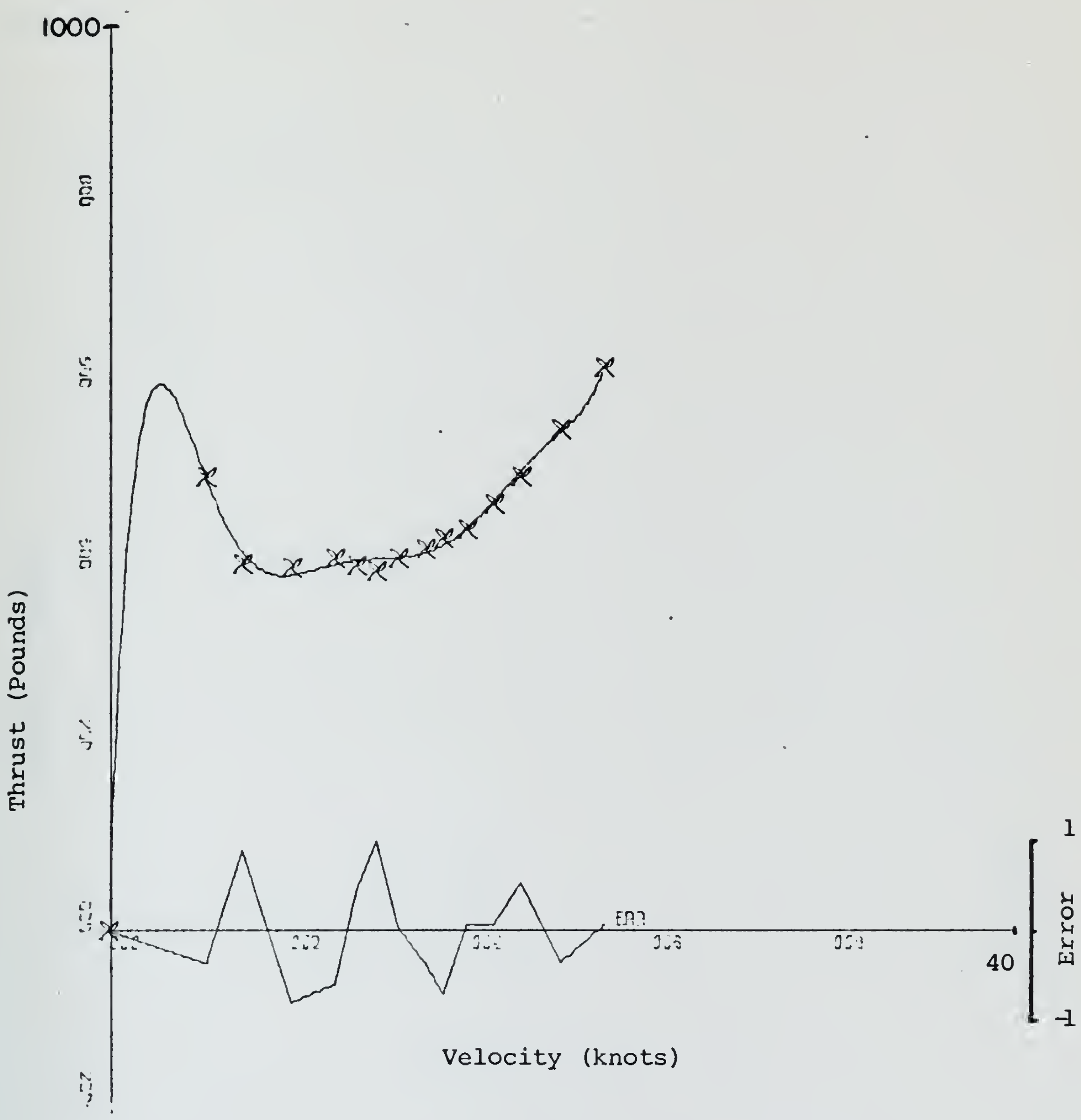
Velocity vs Thrust - 5th Order Fit

Figure 17. Velocity vs Thrust Curves.



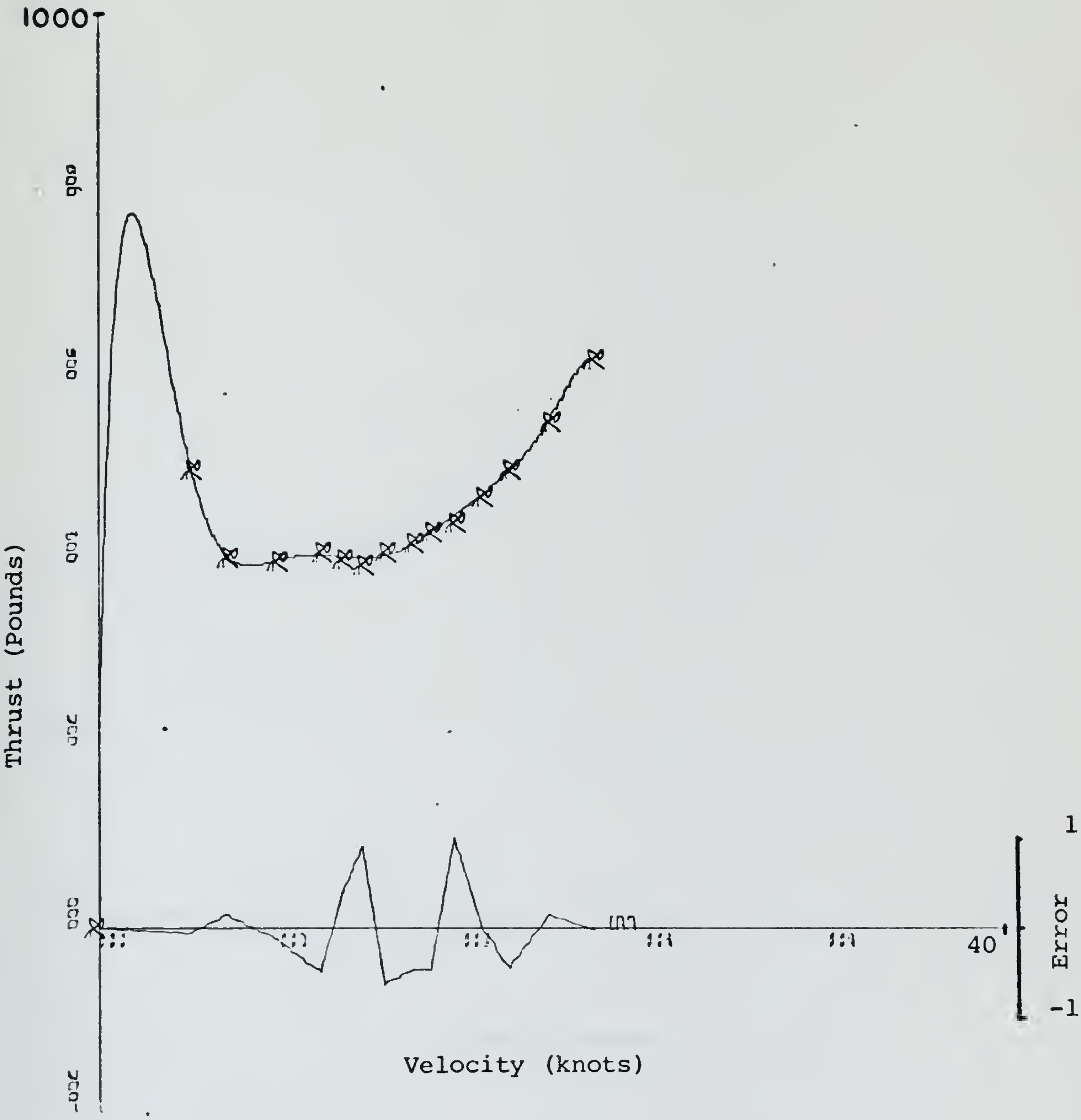
Velocity vs Thrust - 6th Order Fit

Figure 17. (Continued)



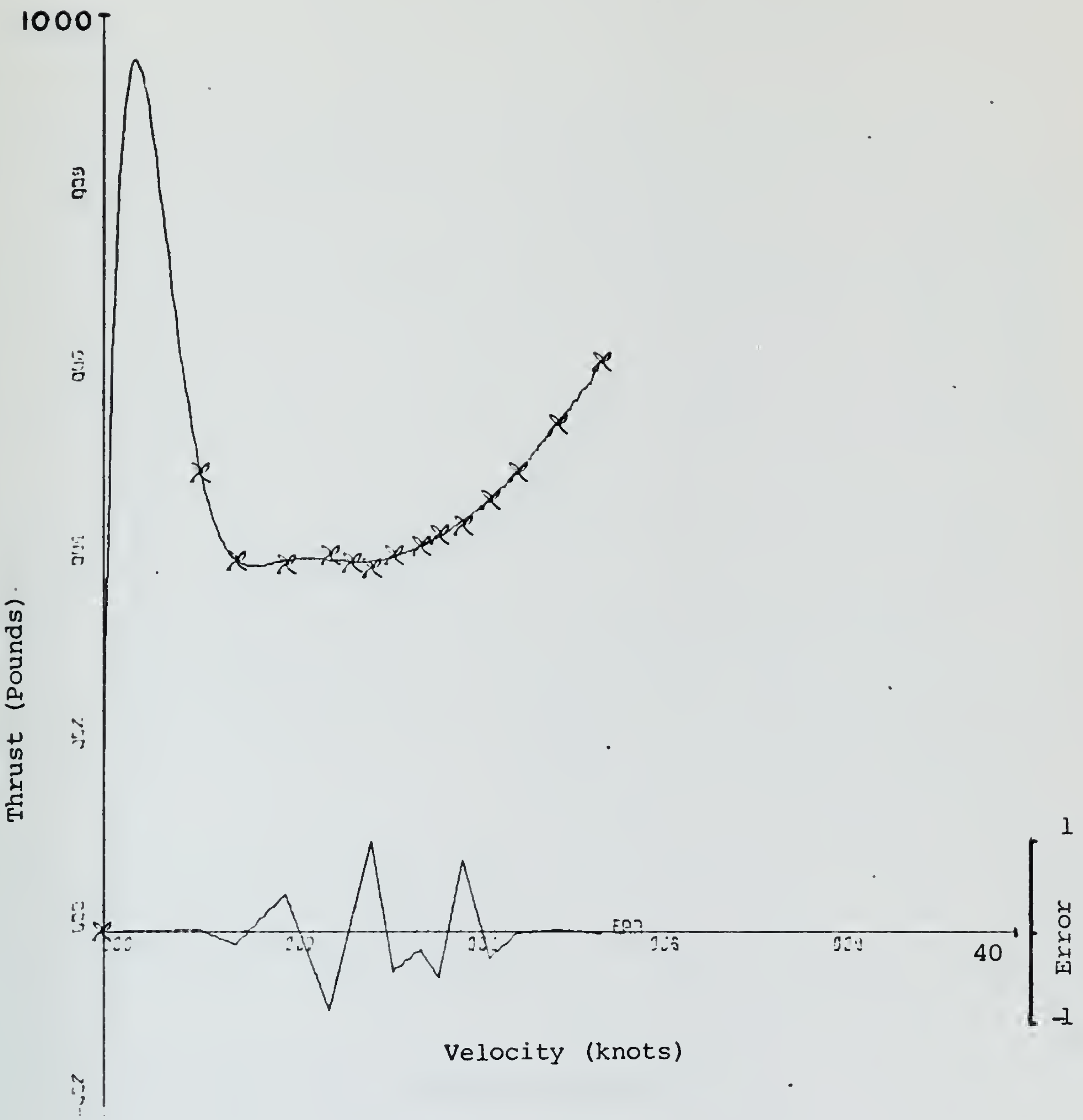
Velocity vs Thrust - 7th Order Fit

Figure 17. (Continued)



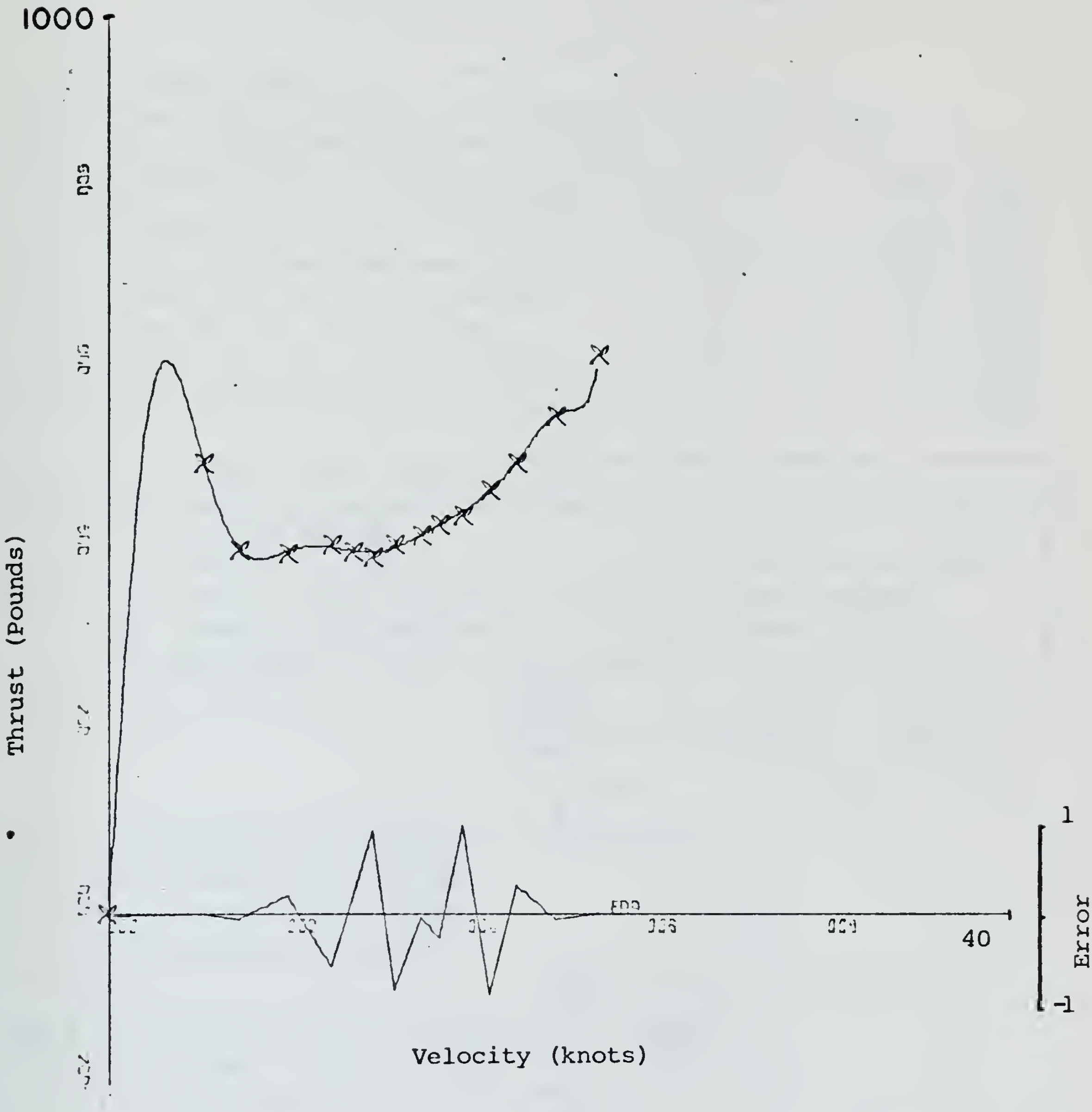
Velocity vs Thrust - 8th Order Fit

Figure 17. (Continued)



Velocity vs Thrust - 9th Order Fit

Figure 17. (Continued)

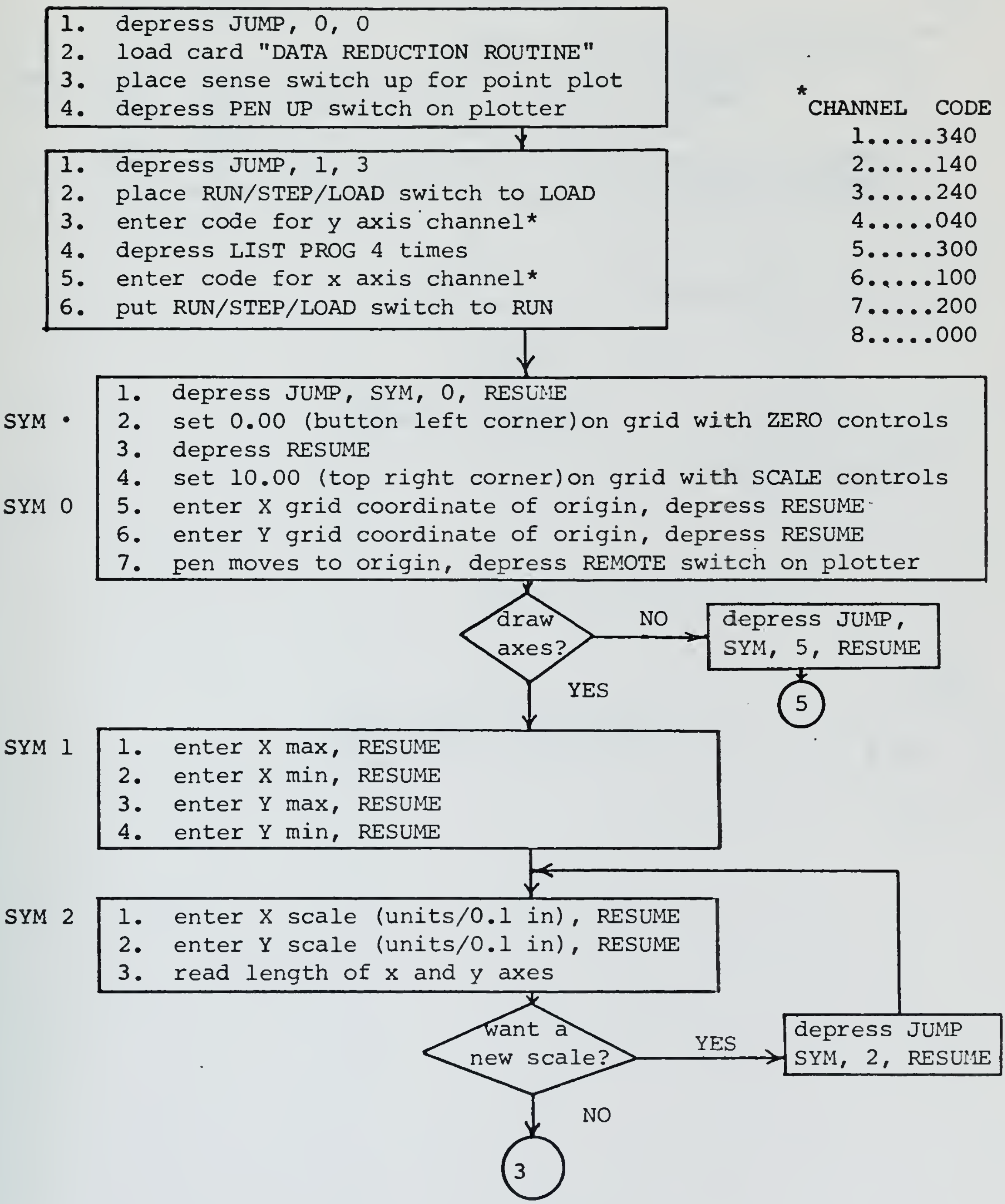


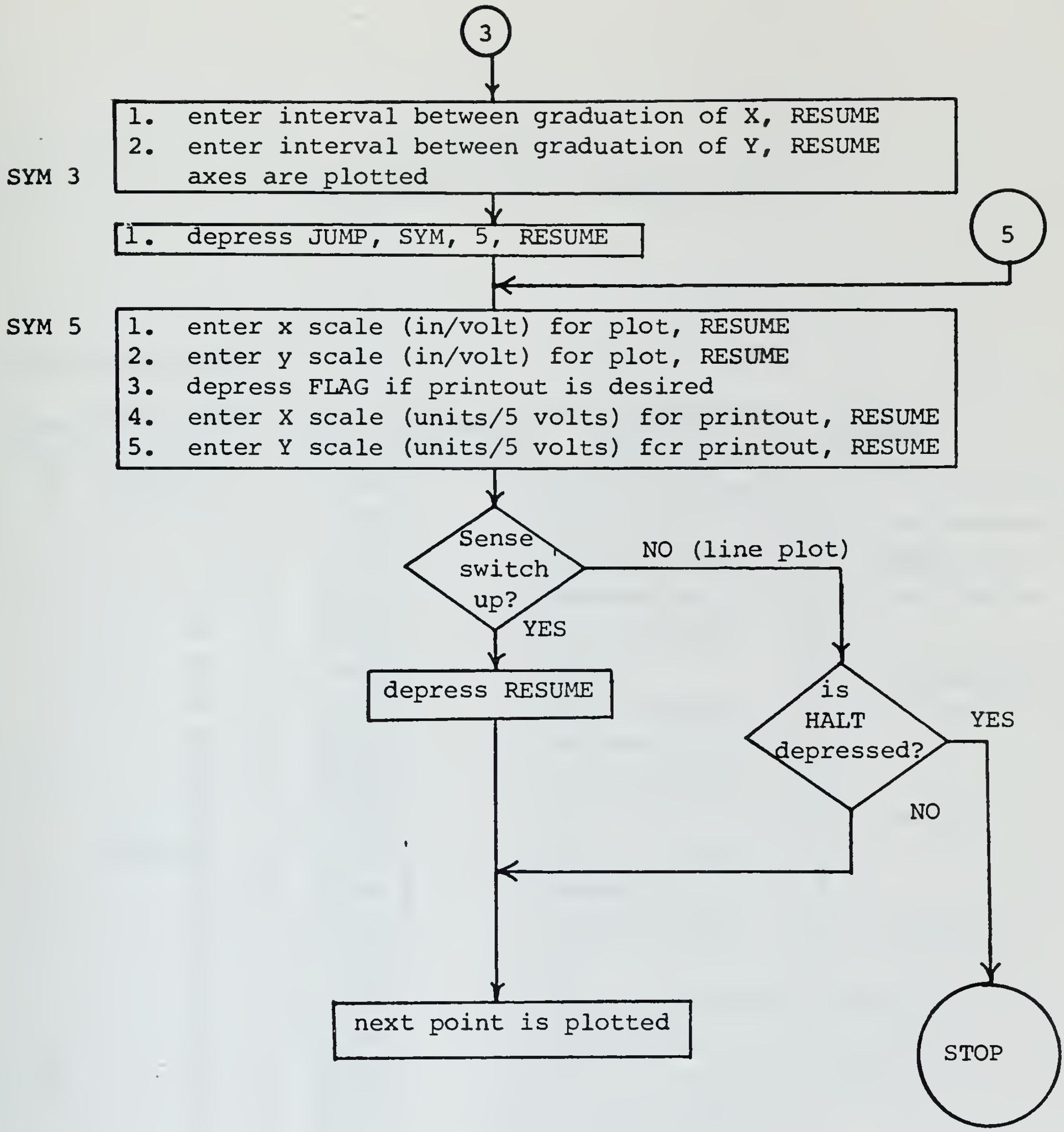
Velocity vs Thrust - 10th Order Fit

Figure 17. (Continued)

APPENDIX C

FLOW CHART FOR DATA REDUCTION PROGRAM




3
1. enter interval between graduation of X, RESUME
2. enter interval between graduation of Y, RESUME
SYM 3
axes are plotted
1. depress JUMP, SYM, 5, RESUME
5
SYM 5
1. enter x scale (in/volt) for plot, RESUME
2. enter y scale (in/volt) for plot, RESUME
3. depress FLAG if printout is desired
4. enter X scale (units/5 volts) for printout, RESUME
5. enter Y scale (units/5 volts) fcr printout, RESUME
Sense switch up?
NO (line plot)
YES
depress RESUME
is HALT depressed?
YES
NO
next point is plotted
STOP

APPENDIX D

DATA REDUCTION PROGRAM LISTING

Statement Number

Octal	Decimal	Code	Symbol	Explanation (if needed)	
0	0000	266		load IX register	
1	1	160		with 160	to transfer
2	2	220		jump to octal	data
-3	3	016		address 16	to plotter
4	4	266		load IX with	
5	5	161		161	to transfer
6	6	220		jump to octal	Y data
7	7	016		address 16	
10	8	266		load IX with	
1	9	162		162	pen up
2	0010	220		jump to octal	
3	1	016		address 16	
4	2	266		load IX with	pen down
5	3	163		163	
6	4	160		store IX into IRO register	
7	5	314		OPCB (output control byte)	
20	6	260		if IL of IX $\neq$ 0 jump	
1	7	025	α^x	to octal address 25	
2	8	314		OPCB	
3	9	220		jump to octal	
4	0020	034	n	address 34	
5	1	302		RINS (read input status)	
6	2	247		if H8 of IX = 1, jump to	
7	3	025	α^x	octal address 25	
30	4	161		recall IRO register to IX	
1	5	314		OPCB	
2	6	241		if L2 of IX = 1, jump to	
3	7	051	[illegible]	octal address 51	
4	8	360		clear IX	
5	9	326		transfer digit of E to IL	
6	0030	315		OPDB (sign transfer)	
7	1	273		load DH with 111 111	

40	0032	277		load DL with 011 111 111
1	3	322		load IX with (DC)*
2	4	315		OPDB (transfer exponents)
3	5	336		decrement DC by 1 step
4	6	322		load IX with (DC)*
5	7	315		OPDB (transfer first 2 digits)
6	8	336		decrement DC by 1 step
7	9	322		load IX with (DC)*
50	0040	315		OPDB (transfer next 2 digits)
1	1	057		RESUME (end plot subroutine)
2	2	066		symbolic 52: subroutine to
3	3	052	F	connect binary to decimal
4	4	114	*	clear data register
5	5	000	0	0
6	6	063	a	clear entry register
7	7	377		no operation
60	8	240		jump if L1 = 1 to
1	9	120	↓	octal addres 120
2	0050	241		jump if L2 = 1 to
3	1	127	&	octal address 127
4	2	242		jump if L4 = 1 to
5	3	135		octal address 135
6	4	243		jump if L8 = 1 to
7	5	143		octal address 143
70	6	244		jump if H1 = 1 to
1	7	151		octal address 151
2	8	245		jump if H2 = 1 to
3	9	156		octal address 155
4	0060	246		jump if H4 = 1 to
5	1	163		octal address 163
6	2	247		jump if H8 = 1 to
7	3	170		octal address 170
100	4	111	↑	
1	5	000	0	
2	6	024	+	to reconvert
3	7	002	2	input data
4	8	005	5	E register now
5	9	006	6	contains input
6	0070	020	=	data value
7	1	057		RESUME

110	0072	377			
1	3	377			
2	4	377			no operations
3	5	377			
4	6	377			
5	7	377			
6	8	377			
7	9	377			
120	0080	001		1	add 128 to main data register (MDR)
1	1	002		2	0 if most significant
2	2	010		8	but (L1) of IX register
-3	3	113		+	is on (L1 = 1)
4	4	000		0	
5	5	220			jump to octal
6	6	062	Λ		address 62
7	7	006		6	add 64 to MDR 0
130	8	004		4	if L2 = 1
1	9	113		+	
2	0090	000		0	
3	1	220			jump to octal
4	2	064			address 64
5	3	003		3	
6	4	002		2	add 32 to MDR 0
7	5	113		+	if L4 = 1
140	6	000		0	
1	7	220			jump to octal
2	8	066			address 66
3	9	001		1	
4	0100	006		6	add 16 to MDR 0
5	1	113		+	if L8 = 1
6	2	000		0	
7	3	220			
150	4	070	S		jump to octal
1	5	010		8	address 70
2	6	113		+	add 8 to MDR 0
3	7	000		0	if H1 = 1
4	8	220			jump to octal
5	9	072	R°		address 72
6	0110	004		4	
7	1	113		+	add 4 to MDR 0

160	0112	000	0	if H2 = 1
1	3	220		jump to octal
2	4	074	r	address 74
3	5	002	2	add 2 to MDR 0
4	6	113	+	if H4 = 1
5	7	000	0	
6	8	220		jump to octal
7	9	076	LR	address 76
170	0120	001	/	add 1 to MDR 0
1	1	113	+	if H8 = 1
2	2	000	0	
3	3	220		jump to octal
4	4	100	P	address 100
5	5	377		
6	6	377		no operations
7	7	377		next is channel select and
200	8	377		data transfer code for A/D converter
1	9	266		load IX with
2	0130	000	0	(code for Y data channel)
3	1	314		OPCB)get y value in IX
4	2	305		IPDB)
5	3	163		store IX in IR1 register
6	4	266		load IX with
7	5	000	0	(code for x data channel)
210	6	314		OPCB)get x value in IX
1	7	305		IPDB)
2	8	057		RESUME
3	9	066		symbolic 75 point
4	0140	075	Z	plot subroutine
5	1	200		branch to octal
6	2	014		address 14
7	3	200		branch to octal
220	4	010		address 10
1	5	056		halt
2	6	057		RESUME
3	7	066		symbolic sin x data scaling
4	8	070	S	and transfer subroutine
5	9	023	X	
6	0150	111	↑	EX MDR 2 for x scale
7	1	002	2	

230	0152	021	+	
1	3	111	↑	add x origin
2	4	010	8	
3	5	020	=	
4	6	220		jump to x data entry of
5	7	000	0	plot routine
6	8	066		symbolic arc sine Y data scaling
7	9	071	5	and transfer subroutine
240	0160	023	X	
1	1	111	↑	E X MDR 3 for Y scale
2	2	003	3	
3	3	021	+	
4	4	111	↑	add Y origin
5	5	011	9	
6	6	020	=	
7	7	220		jump to Y data entry
250	8	004	4	plot routine
1	9	066		symbolic 5 scale factors
2	0170	005	5	storing subroutine
3	1	065		advance paper
4	2	166		clear flag
5	3	117		set decimal point to
6	4	003	3	3 places
7	5	200		jump to pen up entry
260	6	010	8	of plot routine
1	7	056		halt
2	8	110	↓	
3	9	002	2	enter and print x scale
4	0180	060		for plot
5	1	056		halt
6	2	110	↓	enter and print x scale
7	3	003	3	for printout
270	4	060		
1	5	056		halt
2	6	110	↓	enter and print Y scale
3	7	004	4	for plot
4	8	060		
5	9	056		halt
6	0190	110	↓	enter and print Y scale
7	1	005	5	

300	0192	060		for printout
1	3	065		advance paper
2	4	066		symbolic 6
3	5	006	6	main routine
4	6	200		branch to octal address 201
5	7	201		get x and y data
6	8	200		branch to octal address 54
7	9	054	1/x	evaluate x data
310	0200	127	Br	branch if flag is set to
1	1	016		symbolic r(x print
2	2	067		routine
3	3	074	r	
4	4	127	Br	branch to symbolic sin
5	5	067		(x scale and plot routine)
6	6	070	S	
7	7	165		recall IR1 register to IX(Y data)
320	8	200		
1	9	054	1/x	evaluate Y data
2	0210	127	Br	branch of flat is set to
3	1	016		symbolic 1/x (Y print
4	2	067		routine)
5	3	054	1/x	
6	4	127	Br	branch to y scale and
7	5	067		plot routine
330	6	071	Ŝ	
1	7	127	Br	
2	8	023	X	
3	9	067		
4	0220	075	z	
5	1	126	Ju	jump to symbolic 6 again
6	2	067		(start of main program)
7	3	006	6	
340	4	066		symbolic r
1	5	074	r	x print routine
2	6	110	↓	
3	7	001	/	save x value
4	8	111	↑	
5	9	023	X	
6	0230	004	4	scale x
7	1	266		load IX with

350	0232	354		11 101 100
1	3	150		print scaled x and x
2	4	111	↑	
3	5	001	/	return x value
4	6	057		RESUME
5	7	066		symbolic 1/x
6	8	054	1/x	Y print routine
7	9	110	↓	
360	0240	001	/	same Y value
1	1	111	↑	
2	2	023	X	
3	3	005	5	scale Y
4	4	266		load IX with
5	5	355		11 101 101
6	6	150		print scaled Y and Y
7	7	176		print data
370	8	111	↑	
1	9	001	/	return Y value
2	0250	057		RESUME
3	1	066		symbolic • plotter "zero"
4	2	012		and scale check routine
5	3	062	∧	
6	4	060		
7	5	200		
400	6	000	0	send 0 to plotter
1	7	200		
2	8	004	4	
3	9	056		
4	0260	001	/	
5	1	000	0	send 10 to plotter
6	2	060		
7	3	200		
410	4	000	0	
1	5	200		
2	6	004	4	
3	7	066		symbolic 0
4	8	000	0	grid coordinate routine
5	9	056		
6	0270	110	↓	
7	1	010	8	enter X grid coordinate

420	0272	060	.	
1	3	200		
2	4	000	0	
3	5	056		enter Y grid coordinate
4	6	110	↓	
5	7	011	9	
6	8	060		
7	9	200		
430	0280	004	4	
1	1	066		symbolic 1 axis plot
2	2	001	1	routine
3	3	056		enter x maximum
4	4	110	↓	
5	5	000	0	
6	6	060		
7	7	056		enter x minimum
440	8	110	↓	
1	9	001	1	
2	0290	060		
3	1	056		enter y maximum
4	2	060		
5	3	110	↓	
6	4	004	4	
7	5	056		enter y minimum
450	6	060		
1	7	110	↓	
2	8	005	5	
3	9	066		symbolic 2
4	0300	002	2	
5	1	056		enter x scale
6	2	060		
7	3	110	↓	
460	4	002	2	
1	5	056		enter y scale
2	6	060		
3	7	110	↓	
4	8	003	3	
5	9	111	↑	
6	0310	000	0	
7	1	111	↑	

470	0312	022	-	
1	3	001	1	
2	4	353		
3	5	111	↑	
4	6	024	÷	
5	7	002	2	
6	8	060		compute and print length
7	9	111	↑	of x axis
500	0320	004	4	
1	1	111	↑	
2	2	022	-	
3	3	005	5	
4	4	353		
5	5	111	↑	
6	6	024	÷	
7	7	003	3	
510	8	060		compute and print length
1	9	056		of y axis
2	0330	060		enter interval between graduations
3	1	110	↓	of x
4	2	006	6	
5	3	056		enter interval between
6	4	060		graduations of y
7	5	110	↓	
520	6	007	7	
1	7	066		symbolic 3
2	8	003	3	axis plot routine
3	9	062	∧	
4	0340	111	↑	
5	1	001	1	
6	2	041	+	
7	3	127	&	
530	4	067		
1	5	070	S	transfer x value
2	6	000	0	
3	7	127	&	
4	8	067		
5	9	071	S	transfer y value
6	0350	127	&	
7	1	067		

540	0352	077	SD	
1	3	066		symbolic lg
2	4	050	lg	routine to plot x axis
3	5	043	◇	
4	6	127	Br	
5	7	067		
6	8	070	S	
7	9	000	0	
550	0360	127	Br	
1	1	067		
2	2	071	$\bar{S}$	
3	3	043	◇	
4	4	127	Br	
5	5	067		
6	6	070	S	
7	7	012		
560	8	001	/	
1	9	111	↑	
2	0370	021	+	
3	1	011	9	
4	2	200		
5	3	004	4	
6	4	043	◇	
7	5	127	Br	
570	6	067		
1	7	070	S	
2	8	012		
3	9	001	/	
4	0380	111	↑	
5	1	022	-	
6	2	011	9	
7	3	013	-	
600	4	200		
1	5	004	4	
2	6	043	◇	
3	7	127	Br	
4	8	067		
5	9	070	S	
6	0390	000	0	
7	1	127	Br	

610	0392	067		
1	3	071	$\bar{S}$	
2	4	111	↑	
3	5	006	6	
4	6	041	+	
5	7	111	↑	
6	8	000	0	
7	9	022	−	
620	0400	043	◇	
1	1	020	=	
2	2	126	Ju	
3	3	021	+	
4	4	067		
5	5	050	lg	
6	6	126	Ju	
7	7	020	=	
630	8	067		
1	9	050	lg	
2	0410	200		
3	1	010	8	
4	2	000	0	
5	3	127	Br	
6	4	067		
7	5	070	S	
640	6	062	∧	
1	7	111	↑	
2	8	005	5	
3	9	041	+	
4	0420	127	Br	
5	1	067		
6	2	071	$\bar{S}$	
7	3	127	Br	
650	4	067		
1	5	077	SU	
2	6	066		symbolic e^x
3	7	051	Mf	y axis plot routine
4	8	000	0	
5	9	127	Br	
6	0430	067		
7	1	070	S	

660	0432	043	◇
1	3	127	Br
2	4	067	
3	5	071	$\bar{S}$
4	6	012	
5	7	001	1
6	8	111	↑
7	9	021	+
670	0440	010	8
1	1	200	
2	2	000	0
3	3	043	◇
4	4	127	Br
5	5	067	
6	6	071	$\bar{S}$
7	7	012	
700	8	001	1
1	9	111	↑
2	0450	022	-
3	1	010	8
4	2	013	-
5	3	200	
6	4	000	0
7	5	043	◇
710	6	127	Br
1	7	067	
2	8	071	$\bar{S}$
3	9	000	0
4	0460	127	Br
5	1	067	
6	2	070	S.
7	3	043	◇
720	4	127	Br
1	5	067	
2	6	071	$\bar{S}$
3	7	111	↑
4	8	007	7
5	9	041	+
6	0470	111	↑
7	1	004	4

730	0472	022	−	
1	3	043	◇	
2	4	020	=	
3	5	126	Ju	
4	6	021	+	
5	7	067		
6	8	051	[illegible]	
7	9	126	Ju	
740	0480	020	=	
1	1	067		
2	2	051	[illegible]	
3	3	062	∧	
4	4	200		
5	5	010	8	
6	6	126	Ju	
7	7	067		
750	8	004	4	
1	9	066		symbolic 77
2	0490	077	SD	routine to allow time for
3	1	000	0	pen to return to origin
4	2	051	[illegible]	after plotting x axis
5	3	050	[illegible]	then lower pen
6	4	051	[illegible]	
7	5	050	[illegible]	
760	6	200		
1	7	014		
2	8	057		

*(DC) indicates the contents of address specified by DC

TABLE VII

DATA REDUCTION PROGRAM SCRATCH PAD REGISTER USAGE

REGISTER	USAGE	
	AXES DRAWING SECTION	PLOTTING SECTION
0	X maximum	decimal conversion
1	X minimum	temporary storage
2	X scale (units/0.1 in)	X scale (V/in)
3	Y scale (units/0.1 in)	Y scale (V/in)
4	Y maximum	X scale (units/V)
5	Y minimum	Y scale (units/V)
6	X interval	not used
7	Y interval	not used
8	grid origin X	grid origin X
9	grid origin Y	grid origin Y

LIST OF REFERENCES

1. Feeney, James Leo, XR-3 Data Acquisition System for Performance Evaluation, M.S. Thesis, Naval Postgraduate School, Monterey, March, 1972.

2. Conte, S. D. and deBoor, C., Elementary Numerical Analysis, 2nd ed., McGraw-Hill, 1972.

3. Naval Underwater Systems Center Report 4059, Difference Method of Fitting Polynomials to Empirical Data, by James B. Donald, 20 March 1971.

4. Ames, W. F., Numerical Methods for Partial Differential Equations, Barnes and Noble, 1969.

5. Millman, J. and Halkiss, C. C., Integrated Electronics, McGraw-Hill, 1972.

6. Monroe Calculator Company, Operating Instructions, Model 1880 Scientific Programmable Printing Calculator, 1972.

7. Monroe Calculator Company, Programming Reference Manual, Model 1880 Scientific Programmable Printing Calculator, 1972.

8. Monroe Calculator Company, Advanced Programming Reference Manual for Monroe 1800 Models, 1973.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Assistant Professor D. M. Layton, Code 57 Department of Aeronautics Naval Postgraduate School Monterey, California 93940	5
4.	Lieutenant Peter Vincent Dabbieri, Jr., USN 100 Malloway Lane Monterey, California 93940	1
5.	Chairman, Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) Data Reduction System for the XR-3 Captured Air Bubble Testcraft	5. TYPE OF REPORT & PERIOD COVERED Master's Thesis December 1973
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Peter Vincent Dabbieri, Jr.	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940	12. REPORT DATE December 1973
	13. NUMBER OF PAGES 98
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Naval Postgraduate School Monterey, California 93940	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Data Acquisition
Data Reduction
Analog/Digital Conversion
Signal Processing

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A fast, accurate and portable data reduction system was developed for the XR-3 Captured Air Bubble testcraft being evaluated at the Naval Postgraduate School, Monterey, California. The system consists of four units. A magnetic tape recorder is used for data play back. A signal conditioner unit with a built-in analog-to-digital converter was developed and is used to filter, amplify, sum and further prepare the data for

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
(Page 1) S/N 0102-014-6601 |

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

transmission to either a strip chart recorder or a digital X-Y plotter through a Monroe 1880 calculator.

Preliminary use of curve fitting techniques are discussed; calculator programming and the various problems and solutions encountered in the development of the system are described.

140027

Thesis
D1214
c.1

Dabbieri
Data reduction system for the XR-3 Captured Air Bubble testcraft.